STRUCTURES AND TYPES OF DIFFERENTIATED STREET GRIDS: THE GENERATION, ANALYSIS, AND SORTING OF UNIVERSES OF SUPERBLOCK DESIGNS

A Dissertation
Presented to
The Academic Faculty

By

Chen Feng

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in Architecture

Georgia Institute of Technology

August 2019

STRUCTURES AND TYPES OF DIFFERENTIATED STREET GRIDS: THE GENERATION, ANALYSIS, AND SORTING OF UNIVERSES OF SUPERBLOCK DESIGNS

Dr. John Peponis, Advisor
School of Architecture
Georgia Institute of Technology

Dr. Ruth Conroy Dalton
Department of Architecture and Built Environment
Northumbria University

Dr. Sonit Bafna
School of Architecture
Georgia Institute of Technology

Dr. José Pinto Duarte
Stuckeman School of Architecture and Landscape Architecture
Pennsylvania State University

Dr. Jarek Rossignac
School of Interactive Computing
Georgia Institute of Technology

Dr. Andres Sevtsuk
Graduate School of Design
Harvard University

Date Approved: May 17, 2019

ACKNOWLEDGEMENTS

First and foremost, I wish to thank my advisor, John Peponis, for his artful guidance and for being a model scholar. His way of thinking and seeing architecture has profoundly influenced my own understanding of architecture—and life. I enjoyed the delightful conversations with Sonit Bafna who provided both deep insights and fresh perspectives. Jarek Rossignac introduced me to the concepts and methods originated in the field of computer graphics that are related to my study. The critical comments and questions from Ruth Dalton, José Duarte, and Andres Sevtsuk at my defense have truly helped me clarify the vague points and strengthen the key arguments in the thesis. In addition to those who directly contributed to this dissertation, I am grateful for the financial support received from George Johnston, Scott Marble, and Julie Kim over the years of study at Georgia Tech. I appreciate the teaching opportunities provided by Frederic Pearsall and Laura Hollengreen. The walkability study that I did with Ellen Dunham-Jones helped me to know the issues raised by the design of American cities better. Athanassios Economou introduced me to Pólya enumeration theorem which proved useful in my research. Robin Tucker capably guided me through all administrative processes and made them seem natural and easy. I also enjoyed the company of and collaboration with Martin Scoppa, Alice Vialard, Julie Zook, and Dawn Haynie, who were doctoral students working in the same research field. Qinpeng Wang and Zhichen Xia, both close friends, shared with me the happy and sad moments. My final thanks are to my parents, Hongjuan He and Ke Feng, for their unconditional love, and my wife, (soon to complete her doctoral thesis) Dan Sun, for her continuous encouragement, patience, and optimism. At this moment I also remember with gratitude

Haofeng Wang who provided the first foundations for my work in space syntax and guided my earlier research prior to my joining the doctoral program at Georgia Tech while also continuing to exchange ideas and work on common projects after.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

α	the threshold angle used to determine a change of direction
dc	direction change
DCEL	doubly connected edge list
DDL	directional distance per length
DDL10d	directional distance per length with the threshold angle set to 10 degrees
DDL20d	directional distance per length with the threshold angle set to 20 degrees
DDL30d	directional distance per length with the threshold angle set to 30 degrees
DDL35d	directional distance per length with the threshold angle set to 35 degrees
dr0dc20d	zero-direction-change reach with the threshold angle set to 20 degrees
dr2dc20d	two-direction-change reach with the threshold angle set to 20 degrees
sd	standard deviation
T	the number of times to apply a specific operation

SUMMARY

The design of urban street networks is critical to how a city looks, feels, and functions. Moreover, the arrangement of streets inside the “superblocks”, which are the large urban areas divided up by the primary street network of the city, gives cities unique characters. This dissertation studies the street network designs at the scale of a square superblock that measures half a mile, or 800 m, on each side—a particularly common dimension for the spacing of arterial streets in the U.S., China, and many other countries.

The contemporary urban landscape has been significantly shaped by two distinctive traditions for organizing streets at the scale of a superblock. At one extreme is the deployment of a uniform grid, differentiated only by street widths or intensity of development along the streets. At the other extreme is the “tree-like” pattern in which most separate branches or disjoined enclaves or loops are attached to the main streets, imposing a segregating hierarchy defined by mobility and access. This study explores street network designs that fall between these extremes; the designs in question can be described as *differentiated grids*. More specifically, we ask: (a) How to create differentiated grids by progressively deforming a square grid? (b) What different kinds of differentiated grids are there? (c) What is the relationship between the different rules that can be applied to creating differentiated grids and the emerging types of differentiation?

To study those questions, eight different “syntactic operators” have been developed to progressively deform a street network. For each type of operation, a generative rule/algorithm was created to sequentially apply the operation on a uniform grid up to a specified number of times. An additional generative algorithm was also

created to allow operations to be mixed in random sequences. Each generative algorithm was applied to generate a total of 600 differentiated street grids. This resulted in a "design universe" consisting of 5400 differentiated street grids that could be analyzed comparatively and queried for the presence of properties of interest. Such properties include graph connectivity, street density, block size and shape, intersection density, geometric regularity, directional reach, directional distance, and the diversity in syntactic conditions.

In addition, the centrality structure of designs was studied. The aim was to formulate and test alternative definitions of "integration cores" and to develop relevant typologies. Consistent with space syntax literature, an integration core is defined as comprising the streets that are closer to all parts of the street network in terms of directional distance. Query algorithms were developed to select designs based on the definitions of alternative types of integration cores.

Four main conclusions were reached. First, different types of operations have different capacities to influence the properties of a street network. Second, there are multiple dimensions of differentiation (e.g., differentiation in geometric alignment of streets, differentiation in configurational properties such as DDL, differentiation in block shapes, etc.). In many cases, measures along the different dimensions of differentiation are related. Their *predictable* relationship can be quantified. Third, while the relationship between different dimensions of differentiation usually has a consistent direction, its slope can vary, depending on the type of operation used to create the differentiation. The variation in slope suggests that properties that may be desirable (for example the creation of a diversified street grid) can be achieved with varying costs regarding properties that

may be undesirable (for example the creation of less accessible streets). Fourth, the (*local*) generative rules used to generate designs do *not* necessarily lead to specific emergent global properties of the street network of the superblock. Although we cannot *predict* the specific syntactic type we get by applying a specific generative rule, we know that by applying certain generative rules, we are *more likely* to generate designs of a specific syntactic type.

Thus, the thesis makes two significant contributions to the field of space syntax studies. First, it demonstrates how the systematic generation and querying of universes of designs can be used to rigorously define and enrich key syntactic ideas that have hitherto remained intuitive, such as the ideas of "deformed grid" and the "shape of the integration core". Second, it demonstrates that in principle, the design of street networks at superblock scale can be studied according to the typologies of interface between local and global integration and according to the typologies of differentiation of the street grid.

CHAPTER 1
INTRODUCTION

1.1 Research Background

1.1.1 The design of street networks

The design of streets is critical to how a city looks, feels, and functions. Admittedly, there are many aspects pertaining to street design. Some are related to traffic control, such as the designated uses and widths of lanes. Some are related to the design of physical elements, such as lighting, signage, or pavement. However, a particularly important aspect of street design is how they are aligned and connected to each other to form a street network. A street network is an abstraction of the spatial system formed by the publicly accessible open spaces in the city which "carries different speeds of movement and different rates of perception", and as such, it contributes to the total living experience in the city and serves "as a dominant organizing force in architectural design" (Bacon, 1976, pp. 34–35). Careful design and planning of street networks is not only important for the easy, efficient and safe movement of people and goods but also for supporting different types and scales of development, easing navigation and encouraging exploration, facilitating frequent and diverse social interactions, and fostering vibrant local cultures—that is, an urbanism of pluralism and openness (Jacobs, 1961; Peponis, 2006; Peponis & Feng, 2016; Peponis, Park, & Feng, 2016).

1.1.2 The types of street networks

When we approach the design of street networks in the field of architecture, identifying different types of street networks becomes critical as the design process is arguably "a way of bringing the elements of a typology—the idea of a formal structure—into the precise state that characterizes the single work" (Moneo, 1978, p. 23). The concept of type is central to architecture as a profession and as a discipline. As Moneo (1978) pointed out, Architecture "is not only *described* by types, it is also *produced* through them" (p. 23). Hence, while street networks have been studied from various perspectives across multiple disciplines[1], of particular interest to architects and urban designers is the study of types of street networks.

Conceptually, it is impossible to enumerate all types of street networks: we can bring in arbitrarily many descriptors (qualitative or quantitative) and be arbitrarily precise in characterizing street networks. In other words, there are always new ways of grouping objects and new ranks of types. Even if we concede that each street network embodies a

[1] The study of street networks has drawn attention from scholars in various disciplines, including architectural historians, urban designers, city planners, transportation engineers, geographers, environmental psychologists, cognitive scientists, and physicists (Barthelemy, 2011; Boeing, 2018; Christova, Scoppa, Peponis, & Georgopoulos, 2012; Courtat, Gloaguen, & Douady, 2011; Crucitti, Latora, & Porta, 2006; Hochmair & Frank, 2000; Javadi et al., 2017; Jiang & Claramunt, 2004; Kirkley, Barbosa, Barthelemy, & Ghoshal, 2018; Kostof, 1991; S. Marshall, Gil, Kropf, Tomko, & Figueiredo, 2018; W. Marshall & Garrick, 2010; Sadalla & Magel, 1980; Snellen, Borgers, & Timmermans, 2002; Strano et al., 2013; Xie & Levinson, 2007). Besides, a substantial body of literature on street network analysis can be found in the field of space syntax, which will be introduced later in this chapter.

unique type on its own, it would be a futile attempt to enumerate all instances of street networks: the street networks—on paper or on ground—are constantly created, destroyed, and transformed. It adds more complexity if we further bring up the issue of scale: a continuous street network can cover an arbitrarily large area yet exhibit different characters at different places and different scales, so should it be treated it as one street network or several street networks combined? For all these reasons, the set of street network types is open and open to change.

Nevertheless, specific types of street networks have been identified, discussed and become well recognized by researchers studying the form and structure of real and conceptual street networks. The study of types of street networks along this line of research often involves visual inspections, spatial analysis, and statistical analysis. The raw data used for such kind of study are usually cadastral maps or street maps. However, the street networks may be modelled differently and represented by different kinds of graphs for advanced spatial analysis[2]. The resulting types are semantically, graphically, or quantitatively described. Below we introduce a few influential pieces of studies that focus on the typology of street network patterns[3].

---

[2] S. Marshall et al. (2018) and S. Marshall (2016) discussed different representational schemes of street networks, including the "primal" graph, the "dual" graph, and the route structure and the line structure representations.

[3] Studies from the field of space syntax are largely omitted in this section but will be introduced later in the chapter after the introduction of basic concepts, measures, and theories of space syntax.

Southworth and Owens (1993) suggested a typology of street forms based on street patterns observed in the metropolitan fringe in the San Francisco Bay area. The street patterns were studied at both the scale of a community and the scale of a neighborhood. At the community scale, based on a sample of eight study areas that measure about nine square miles each (3 × 3 mi., or about 4.8 × 4.8 km), four street patterns were implied based on the manner and extent of their outward and inward growth: (a) speculative gridiron, (b) interrupted parallels, (c) incremental infill, (d) loops and lollipops, and (e) hybrid of gridiron, interrupted parallels, and cul-de-sacs (pp. 273–277). At the neighborhood scale, based on a sample of study areas measure about one hundred acres each (about 40 ha., or 636 × 636 m), five patterns were identified: (a) gridiron, (b) fragmented parallel, (c) warped parallel, (d) loops and lollipops, and (e) lollipops on a stick (p. 280). While the types themselves do not appear to arise from a quantitative analysis but rather be imposed, aggregated measures including linear feet of streets, number of blocks, number of intersections, number of access points, and number of loops and cul-de-sacs suggested clear differences regarding the density, circuity, and connectivity among the selected street patterns.

A dedicated study of street patterns has been made by S. Marshall (2005). Starting with an extensive review of the vocabulary developed to describe different types of streets and street patterns, Marshall pointed out that the existing terminology falls short in accounting for the great variety of street patterns on the ground yet has overabundant descriptors referring to essentially the same general property. In his effort to derive a typology of street patterns, he first made a distinction between “composition” and “configuration”. Composition refers to the physical geometry; configuration refers to

more abstract relations including topology. Then he proposed a taxonomy of patterns (a simple and integrated version and a detailed and elaborated version) based on his study of the types mentioned in the literature. The taxonomy, he argued, reflected the three typological frameworks found in the literature: first, "the typologies which express a tripartite set of grid, radial, and linear"; second, "the bipolar distinction between grid and tree"; third, "the use of an 'other' category to mop up irregular cases" (p. 90).

After arriving at a taxonomy of patterns based on qualitative descriptors, S. Marshall (2005) proceeded to quantify patterns based on the route structure analysis he developed. Three basic properties of each route—"continuity", "connectivity", and "depth"—were measured and used to define types of routes. By averaging these properties for all the routes in a street network and mapping each network onto a triaxial diagram (which he called a "netgram"), a theoretically possible spectrum of street networks was shown (p. 140). Four classes of networks were conveniently defined based on the spectrum, including "tributary", "semi-tributary", "semi-griddy", and "griddy" networks.

A fundamental point made by Marshall is that the kind of street pattern desired by urban designers is not necessarily a perfect grid but a grid-like yet more irregular pattern which he called a "characteristic" pattern (p. 154). The characteristic pattern is one of heterogeneity. The heterogeneity of a street network can be evaluated based on three measures, including "irregularity"—derived by "dividing the number of distinct route types by the total number of routes", "recursivity"—derived by "dividing the number of depths by the number of routes (where the number of depths is simply equivalent to the maximum depth)", and "complexity"—defined as "the number of distinct types of route

present over and above the number of distinct types generated by difference in depth alone" (pp. 147–148). The characteristic street structure, as defined by Marshall, is a semi-griddy street network that has a relatively high degree of complexity and irregularity.

Some recent studies relied on advanced statistical methods to classify the street patterns based on selected quantifiable attributes. By studying the individual street-network increments in the Oporto Metropolitan Area over the last 60 years, Serra, Gil, and Pinho (2016) derived eight typomorphologies of street patterns using the *k*-means clustering analysis. The derived types of street patterns differentiate along different morphological dimensions, including size (i.e., the total road length of each increment), cyclicity (indicated by the number of internal urban blocks created), and connectivity (largely indicated by the number of external connections and the number of internal cul-de-sacs produced). Strano et al. (2013) investigated the structural properties of the street networks of ten European cities. They performed a principal component analysis based on the distributions of centralities and their respective moments and suggested that the presence of physical constraints (e.g., traversed by a river or bordered by a lake) may be a key factor in classifying cities. In an effort to "reach a non-ambiguous, scientific classification" of street patterns, Louf and Barthelemy (2014) characterized the street patterns by characterizing the blocks, which are defined as "the cells of the planar graph formed by streets" (p. 2). Based on measures indicating the distributions of the block areas and shapes in a city, they performed a hierarchical clustering analysis on a dataset containing 131 cities across the world and identified four distinct classes of street patterns: (a) cities that have blocks of medium size with shapes that are dominated by

squares and rectangles; (b) cities mainly made of small blocks with a diversity of shapes; (c) cities that have a diversity of block shapes but are balanced in terms of block areas; and (d) cities made of small, square-shaped blocks.

1.1.3 Grid as a distinctive type of street network

The word *grid*, in a broad sense, suggests a geometric pattern formed by a set of crisscrossed lines or curves as well as the openings enclosed by those lines or curves. It is not always clear what people refer to when they describe a street network as "grid-like". For example, they may only refer to the fact that the openings (interpreted as urban blocks) enclosed by the street network have rectangular or square shapes—even though the streets may be frequently interrupted to form many T-intersections. Alternatively, they may simply refer to the fact that the overall pattern of connectivity of the street network is similar to that of a square grid (i.e., dominated by four-way intersections)—even though the streets may intersect each other at various angles and the streets themselves be curvilinear. The term becomes more ambiguous as we notice that there are other kinds of regular tessellations that may be described as grid-like—for example, the triangular tiling (sometimes called triangular grid) and the hexagonal tiling (sometimes called hexagonal grid). However, grid, as a clearly defined and distinctive type of street network, almost always refers to the street pattern where two sets of parallel lines run across each other at right angles, forming rows and columns of squares or rectangles. As we shall see, this interpretation gains much wider recognition when it comes to urban design and city planning.

1.1.4 Grid as a concept of urban planning

Often discussed in the fields of urban design and city planning, a grid city is a city that has a checkerboard or a gridiron plan in which the streets intersect perpendicularly to form blocks of regular sizes and shapes. In fact, the grid pattern of land subdivision has been adopted as a geometric basis of planned settlements in various places of the world and over thousands of years of human history. Miletus, an ancient Greek city, is probably the best-known example of orthogonal planning in ancient times. The masterplan was prepared by a Milesian architect, Hippodamus of Miletus, who according to Aristotle (trans. 1932) invented the division of cities into blocks and cut up Piraeus. The adoption of grid as an urban form regulator could be traced back to the early third millennium BC Harappan cities and up to the much more recent speculative grid towns of the United States (Kostof, 1991; Morris, 1994).

1.1.5 Uniform grid

What is a uniform grid?

In the context of urban design and city planning, a uniform grid—sometimes called a regular or perfect grid—indicates a street layout in which straight and continuous streets are spaced at equal intervals and cross each other at right angles, forming blocks of the same shape and size. Depending on the actual shape of the blocks dictated by such a grid, it is also known as either the checkerboard (with square blocks) or the gridiron (with oblong blocks) plan. It is the simplest and perhaps also the most rigid kind of grid.

Therefore, uniform grids do not refer to a single fixed pattern but rather a whole category of patterns which meet the described condition. Even in its purest form—that is, devoid of any environmental and socio-economical content—one uniform grid can still differ from another in terms of the size and proportion of the blocks and the spacing of the streets—the fact of which alone could lead to an infinite set of theoretical variations.

Uniform grid in the history of urban forms

Cities originated from orthogonal planning schemes are a common topic in books surveying the historical urban forms (Kostof, 1991; Morris, 1994; Reps, 1965). However, it is important to note that orthogonal planning schemes do not necessarily yield uniform grid patterns—if we define the orthogonal planning simply as the general principle of setting built elements at right angles. To ensure an orthogonal plan to be a uniform grid, at least three additional conditions need to be met: (a) the built elements being set at right angles should refer to the streets—instead of, say, partition walls, or rooflines; (b) every street at the proper size level should be straight and continuous all through the grid; (c) those streets should be spaced at equal intervals, forming blocks of the same shape and size. In that sense, the plan of Suzhou, China, as recorded in the stone engraving prepared in 1229, is indeed orthogonal, but far from a uniform grid; by the same token, the famous town plan of Savannah (also known as the Oglethorpe Plan) may be called a tartan grid, but not a uniform grid.

The plan of Miletus laid out after the liberation of Ionia in 478 B.C. is among the most well-known early examples of a uniform grid city. To allow the fortification walls to follow the most convenient defensive line, the periphery of the city is irregular, as

commonly seen in both planned and unplanned Greek cities (Boardman, 1994). Early Roman grid conforms to the Greek model. Accompanying the Romans' systematic conquest of Italy and the expansion of the empire, the Roman grid developed its own identity. The plan looks more unitary and has large square blocks. The blocks are tightly packed with mural frames that are aligned on the sides facing the streets. The forum is usually placed around the crossing of two main roads; the main north-south street is sometimes called the *cardo* and the main east-west street called the *decumanus*. Timgad, founded in 100 A.D., serves as a perfect example here. Even from these early examples, we can sense a difficulty of consistently applying a uniform grid across the whole city, especially where the size and shape of the building or public space cannot be easily accommodated by the established grid layout. In the case of Timgad, blocks were consolidated to make room for the forum, the theater, and the baths (Kostof, 1991).

Several centuries after the end of the Classical world, the orthogonal planning schemes reappears in Europe from 1100 on, applied both in the creation of new towns—for example, the *bastides* built in France during the Middle Ages—and for the extension of earlier city forms. In colonial America, the grid plan had also been employed as an effective way of planting new towns—for example, New Orleans in New France and New Haven (Connecticut) in New England. The design of towns in New Spain was largely based on the directives emanating from the Spanish court—which were later collected in a document known as the *Laws of the Indies*. In those *pueblos* or *villas*, normally, a large public square is located at the intersection of two main axes, and its size regulated the makeup of the grid (Kostof, 1991, p. 115).

The survey system set up by the National Land Ordinance of 1785 had a profound impact on the shaping of the urban and rural landscape during the westward expansion in the United States after the American Revolution. The vast land lying in the Midwestern and Western U.S. were surveyed into square townships measured 6 by 6 miles, which were further subdivided into sections of one square mile. The sections could again be further subdivided into halves and quarters. The large grid pattern adopted in surveying and distributing the vast rural land in the western territories had undoubtedly inspired the gridded city forms emerged later in those areas, although at a much finer scale.

Despite the prevalence of the gridded towns in the west, the most famous example of a uniform grid in the U.S. lies in the east—the Manhattan grid. In 1811, a three-member commission planned the whole of Manhattan in the form of identical blocks, beginning from the 23rd Street up to the 155th Street. As Kostof (1991) pointed out, the Commissioners' Plan of 1811 signaled a significant shift from the application of the grid in Colonial days.

> Another way of putting it, following Peter Marcuse, is that the 1811 plan of Manhattan represented the abandonment of the Colonial closed grid for the open grid of the new era of the Republic. The closed grid is essentially a pre-capitalist concept. It is seen as having firm boundaries, and a definite design within this fixed frame.…The open grid is predicated on a capitalist economy, and the conversion of land to a commodity to be bought and sold on the market. The grid is left unbounded or unlimited, so it can be extended whenever there is the promise of fast and substantial profit. (p. 121)

The use of a uniform grid—or some less restrained version of it—as a convenient basis to organize the city continued into the 20th century. In certain cases, we see the grid pattern is only revealed at a much larger scale, with superblocks or sometimes called urban sectors locked into the lines of the grid, or, the supergrid. Chandigarh, Islamabad,

and Milton Keynes are good examples. However, it is questionable whether we can still call such cities as grid cities. The grid, formed by freeways and arterial roads, represents only an exclusive set of all the streets in the city. The sub-street systems attached to the supergrid, however, could be wildly different from a grid. Hence it is more accurate to call those cities as cities that have a grid of freeways or arterial roads instead of to call them, indiscriminately, grid cities.

Even by such a brief survey of historical examples of uniform grids, it should become clear that the uniform grid, although seemingly ubiquitous, is varied in terms of its historical origins, the ideological beliefs and social values it expresses, and the technological advances it embodies.

How a uniform grid may be differentiated without changing its geometry?

A uniform grid is often taken as undifferentiated. However, it is rarely true in reality. As soon as it is imposed on the earth, by one way or the other, the "uniform" grid is differentiated, from within or from outside, by human or nonhuman factors.

The grid is, first of all, differentiated by its environmental context. Its constituent parts are differentiated by their distinctive orientation relative to the cardinal points, the sun path, the dominant wind direction, as well as their relationship to the surrounding topography—especially when the grid runs along a hillside or a river bank.

Second, the grid can be differentiated in its internal geometry. For a rectilinear grid, the streets running along the short sides of blocks are different from the rest in terms of the frequency of their intersections with the streets running at right angles to them.

Even within a uniform square grid where streets are placed at equal intervals along both directions, the street width can vary. If we further take the vertical dimension into account, the grid can vary dramatically as seen from the skyline shaped by the buildings rising above the ground–Manhattan being an obvious case.

Third, the grid is not guaranteed to be a socially neutral field. A different location in the grid may imply a different identity and social status—either dictated by systematic planning or driven by the social and economic interactions among individuals and groups historically. This kind of differentiation is evident in the ancient "planned" cities, in which cases the arrangement of a city plan could be associated with cosmological and supernatural symbolism, or deliberate messages about identity and status (Smith, 2007). In a modern capitalist society that embraces free market, different land lots within a grid can have different price tags based on supply and demand. Streets in such a grid may act as dividing lines between different social classes.

Fourth, the grid can be differentiated by regulations on land uses or regulations on traffic circulation. The grid can be treated as a neutral starting point upon which multiple layers of regulation could be imposed to create internal differentiation.

The abovementioned four ways of differentiating a uniform grid (without deforming the grid) are by no means an exhaustive list, but rather to clarify in which senses a uniform grid is neutral and in which it is not—even though it may sound contradictory to its name. The "uniformness" of a uniform grid only refers to the consistency in the spacing of the lines in its pristine diagrammatic form devoid of any realistic context—be it social or environmental. Nevertheless, since our main interest

here is in the geometric forms and structural properties of street networks, we still refer to a street network as a uniform grid or a perfect grid as long as the streets are spaced at equal intervals to form blocks of the same shape and size.

Advantages and disadvantages of a uniform grid

(1) Advantages

There are advantages that have been traditionally associated with a uniform grid in the context of urban design and city planning.

1. It is an effective way to create a sense of orderliness and relatively easy to conceptualize and to implement as a standard scheme for disparate sites.
2. It divides the land evenly and hence facilitates parceling and selling of real estate.
3. It eases surveillance of public spaces by grouping them along straight lines (and because of that, the grid form is also a popular scheme in planning refugee and prisoner camps).
4. It eases navigation around a city since one only needs to reference the distances marked in the two principal directions that the streets run along (although this is frequently aided by a street naming system that indicates the number of blocks between any two streets).
5. It is conducive to the creation of rectangular lots, which tend to bring down the cost in designing and constructing buildings compared to the oddly-shaped lots.
6. It ensures a high degree of street connectivity. With proper block dimensions, it ensures walkability.

Some of those advantages are *scale-independent* and some are *scale-dependent*. For example, the first and second merits of the uniform grid are not inherently about the scale of the grid but rather the simplicity and clarity of the grid as a dimensionless abstract pattern. In this regard, these advantages are as evident in the giant 6-by-6-mile grid of townships resulted from the land ordinance of 1785 as in the much finer, 200-by-200-foot, grid of Downtown Portland. By contrast, the walkability of a uniform grid city is very much dependent on the scale of the grid, or, the side lengths of a block.

Since a uniform grid divides land evenly, it is sometimes treated as a means to achieve egalitarianism. Kostof (1991), however, demystified the common belief that the uniform grid is a symbol of democracy and pointed out that this belief is often not rooted in reality.

> The most persistent belief that urban grids represent an egalitarian system of land distribution is expressed in the context of modern democracies, principally the United States. …The reality is much less admirable. The ordinary citizen gains easy access to urban land only at a preliminary phase, when cheap rural land is being urbanized through rapid laying out. …What matters in the long run is not the mystique of grid geometry, but the luck of first ownership. (p. 100)

(2) Disadvantages

There are also disadvantages that have been traditionally attributed to a uniform grid.

1. It shows no concern to the undulations of the ground, or broadly speaking, the environmental context.

2. It is lack of visual interest[4].
3. It is lack of a clear hierarchy[5].
4. It is not convenient to get to places which are diagonally placed relative to the grid.
5. It consists of cross intersections, which are not considered as safe as T-intersections.

The first disadvantage is about practicality. It points to the wrestle between the formality of the design and the informality of the site. The reconstruction of Priene in the middle of the 4th century B.C. showed such a conflict—streets that ran north-south had to be frequently stepped to negotiate with the sloping site. This is often taken as an example of relentlessly applying a rigid grid system without any concern to the site (Boardman, 1994).

The second disadvantage is an aesthetic one. The trellis-like street network in a uniform grid tends to produce a monotonous effect. As described by Unwin (1994), "the street pictures are not closed, and the vistas wander off in an indefinite, vanishing perspective, often devoid of interest or variety" (p. 235). It does not provide distinctive

[4] It should be noted that some scholars such as Groth (1981) argued that the grid in and of itself does not necessarily lead to monotonous designs—successful streetscape can still be achieved by the skilled planners, designers, and building users.

[5] The lack of hierarchy and visual interest can also make wayfinding difficult since there is considerable visual similarity.

sites for public buildings and it lacks the picturesque quality frequently found in the medieval towns with an irregular plan and curved streets.

The third disadvantage is about the lack of support from the street network itself for the creation or formation of urban centers. It would be impossible to infer the location of the urban centers from the connectivity pattern of the street network alone. Movement in a street system that is in the form of a uniform grid tends to be dispersed everywhere—if there are no variations in the street width, the land use, the traffic regulation, and so on. In that regard, a certain degree of hierarchy or differentiation in the spatial structure of environment may indeed be important for our understanding and appreciation of the city. A hierarchical or differentiated street structure is, in fact, characteristic of the traditional cities that have grown and evolved over a long time, as found by many scholars (Figueiredo & Amorim, 2007; S. Marshall, 2005).

The fourth disadvantage is about the affordance of shortcuts. As Unwin (1994) mentioned, "it does not provide convenient roads for passing to and from the centre, and except when going in two directions, all traffic must travel along two sides of a triangle to get from point to point" (p. 235). Radial and diagonal streets are sometimes introduced to improve the circulation within the gridiron of the street plan. However, it often produces acute-angled plots which are not easy to work with either as building plots or open spaces, as has been criticized by the modern German school of town planners (Unwin, 1994).

The fifth disadvantage is out of concerns on traffic safety. Even before the rise of the automobile and the invention of automated traffic lights, Camilo Sitte advocated the

use of T-intersections by showing with diagrams that it produces the minimum number of collision points. More recent studies also praise the T-intersection for reducing the danger of head-on collisions. For example, Bared and Kaisar (2001) showed that the number of potential conflict locations for a 2×2-lane intersection is reduced from 32 for a cross intersection to 22 when converted to two offset T-intersections. Their study suggests that converting cross intersections to T-intersections are generally desirable in terms of traffic safety in both rural and urban contexts.

In addition to all the disadvantages just mentioned, it is difficult or simply impossible to accommodate certain building types and land uses—such as an airport, a stadium, or a park—with a compact block in a uniform grid. To create a sufficiently large site, several blocks need to be consolidated. In this case, varying a uniform grid is not just preferable but unavoidable.

A choice between a uniform grid and a deformed one is a critical design decision to make. However, the question of whether to design a uniform grid should not be confused with the question of whether to have the related benefits. The reason is simple: the benefits that are related to the uniform grid are not necessarily exclusive to it. Moreover, it is possible to differentiate the grid in such a way that the new arrangement overcomes the shortcomings of the uniform grid while still retain most of the benefits. In fact, more historical urban forms fall into the category of the deformed grid instead of the uniform grid. In this regard, the deformed grid is an interesting topic to pursue. We will study the differentiation of a grid primarily from the perspective of its physical form—more precisely, the differentiation of a street network as represented by street centerlines.

Therefore, in this study, the differentiation of a uniform grid is engendered by deforming the street network.

1.1.6 Deformed grid: a basic definition

While we frequently use the term *deformed grid* in describing and discussing designs of street patterns, our understanding of deformed grids remains largely intuitive. Deformed grids can naturally be thought of patterns resulted from the deformation of a uniform grid. Therefore, deformed grids are clearly different from uniform grids, but still grid-like. However, to be considered as a grid-like structure, a deformed grid, no matter what geometric form it takes, should at least meet the following basic condition—that is, when moving in such a street system, there should (in most cases, if not always) be at least one way to circle back to where one started without traversing the same streets twice. This guarantees that, dead-end streets or cul-de-sacs, if present at all, are the exception. We take this as a basic (the least restrictive) definition of a deformed grid.

1.1.7 Deformed grid in the context of superblock design

What is a superblock?

The term *superblock* has been used in various contexts. Perhaps most straightforwardly, a superblock can refer to a single, unusually large urban block, or a supersized block, especially in comparison of block sizes and forms, following the spirit of the work by Siksna (1998). It can also refer to the large pieces of real estate which are developed holistically and under unified financial control (Colquhoun, 1981). In this study, a superblock is simply defined as the large urban areas bounded by major streets or

arteries. Therefore, based on our definition, a superblock can have internal streets and blocks in the normal sense.

It should be noted that, unlike superblock, the term *block* has clear, established definitions. Although six different definitions associated with the term block are given in the report "A planners dictionary" published by the American Planning Association Planning Advisory Service (Davidson & Dolnick, 2004), they do not vary much from each other. Generally speaking, an urban block refers to a piece of land in the city that is enclosed by publicly accessible streets or a combination of streets, public land, railroad rights-of-way, waterways, or any other physical barriers. In a grid plan, for example, a block is the rectangular area delineated by the orthogonal streets that run along its four sides. In such cases, the blocks simply follow the grid. In the US, there is also a legal dimension attached to the notion of a block, which normally delineates the boundary between the public and private properties.

Superblock, as a planning concept, has been associated with different planning traditions and practices across the world. For example, in the US, the notion of superblocks is an integral part of the neighborhood idea promoted by Stein and Wright and exemplified in their design of the Radburn neighborhood (Patricios, 2002; Stein, 1957). In their design, three or more enclaves (where houses are arrayed in a U-shape about a cul-de-sac court) form a block. Three or four such blocks are grouped around an open green space and form a superblock. Four to six superblocks further form a neighborhood that is bounded by major roads or natural features. In China, the superblocks have been associated with a tradition of cellular urbanism that is exemplified

in both the historical urban fabrics—such as *hutong* (胡同) in Beijing or *lilong* (里弄) in Shanghai—and the modern residential development featured by large bounded neighborhoods (Kan, Forsyth, & Rowe, 2017; Lu, 2006).

Superblocks as important organizing units of urbanism

Cities are powerfully characterized by the arrangement of streets inside the superblocks divided up by the primary street network of the city. Moreover, the superblock has been increasingly conceptualized as an independent design object[6], not in the least boosted by a series of modern planning theories developed in the U.S. and Europe in the early 20th century. Among those, the most famous one is probably the neighborhood-unit concept proposed by the American planner Clarence Perry in 1929. The idea of neighborhood-unit effectively binds the "neighborhood" which is a social concept with an area of the size of a ¼-mile pedestrian shed which is a geo-spatial concept. It has had a large impact on the later residential development tradition in America (Mehaffy, Porta, & Romice, 2015; Perry, 1929).

Meanwhile, the supergrid, which encloses the superblocks, has been increasingly conceptualized as the network of highways and arterial roads for auto-traffic use only. More than often, they become dividers instead of integrators of the city. The concept of functional classification of streets is explicitly stated in the "bible" of traffic engineering,

[6] As Groth (1981) pointed out, "as streetwidth standards increased, so too the size of the units in the urban grid expanded from single fields and other small parcels to whole square-mile units," and "planners with distribution rather concentration in mind used mile-square urban units within their 'maxi-grid' of transportation routes" (p. 75).

sometimes called “the green book”, published by the American Association of State Highway and Transportation Officials (American Association of State Highway and Transportation Officials [AASHTO], 2001).

The concepts and ideas embodied in the model plans or encoded in the technical standards have left an imprint on our contemporary practice. The dimensions of the superblocks have frequently framed the thinking behind urban design projects. In this respect, placing the study of differentiated grids in the context of the common dimensions associated with a superblock is far from arbitrary, but is important for us to understand the contemporary urban landscape. Also, it may inform and enrich our thinking of alternative urban forms.

Superblock design: two extremes and a fuzzy middle range

As superblocks get sufficiently large, say squares of a quarter mile to one mile (or 400–1600 m), there is a wide range of possibilities regarding the organization of the internal street network. We can easily identify two extremes among existing street networks observed at the superblock scale. As shown in Figure 1, at one extreme the superblock resembles a uniform grid, like the one found in Chicago. The grid is only differentiated by street width at the edge. Other than that, it does not offer any form of local area definition or articulation of scale. At the other extreme is the polarized grid, a “tree-like” or “tributary” pattern in which most separate branches or disjoined enclaves or loops are attached to the main streets. It highlights the artificial relationship between mobility and access imposed by the conventional hierarchy of roads advocated by traffic engineers. A superblock found in Los Angeles exemplifies this pattern.

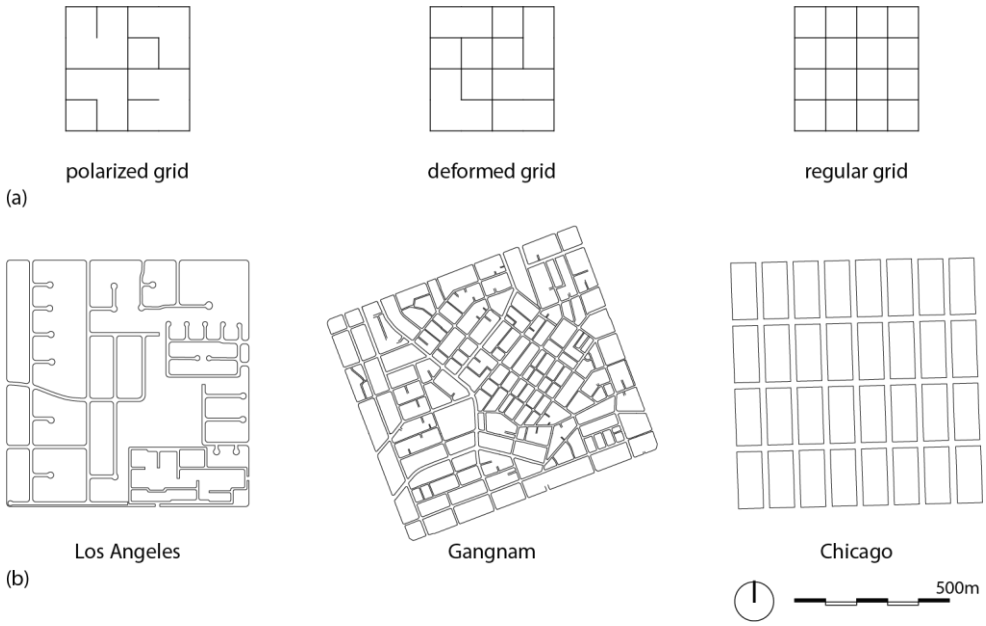

Figure 1. A schematic classification of superblocks: polarized, deformed, and regular grids. (a) Abstract diagrams and (b) actual examples.

Other than the two extremes, the internal street network can sometimes be organized and differentiated in ways analogous to the structure of traditional towns (Peponis et al., 2015). In the particular case of Gangnam, Seoul, we have identified a three-level pattern (Feng & Peponis, 2018; Peponis, Feng, & Park, 2017; Peponis et al., 2016). As shown in Figure 1b, arterial streets are located at the edge of the individual superblock, engaging the city-wide movement. Local main streets traverse the block in one or both directions. The infill local streets are connected up in a way that conforms to our intuitive idea of a deformed grid: some streets are longer, some are more sinuous, some are easier to get to, and some more secluded. However, overall connectivity remains strong. There are no areas of isolation. Cul-de-sacs, if present, are the exception. The deformed grid observed in the superblock in Gangnam can be seen as a particular variation of a regular grid and arguably a good one. However, there are infinite possible

variations along the full spectrum between a perfect/regular grid and a polarized grid. To gain insights into the fuzzy middle range at a conceptual level, we first of all need a precise language to characterize street networks and patterns.

1.1.8 A brief introduction of urban spatial syntax

Space syntax is a descriptive theory of the perceptual and functional affordances of inhabited space that underpin its cognitive and social intelligibility.

The study of urban spatial syntax has largely been based on the analysis and description of street networks as systems of discrete yet interrelated geometric elements such as lines (Hillier & Hanson, 1984; Hillier, Hanson, Peponis, Hudson, & Burdett, 1983). A fundamental concept in space syntax is "universal distance", which takes into account the distance from each element to all others, whether the distance is defined by the number of turns or the angular change required to get from one space to another or the actual path length measured in meters (Hillier, 1996b). The measure of "integration" (defined similarly as "close centrality" in network science), one of the most important measures in space syntax, is a direct function of the universal distance. In urban-scale syntactic analysis, the street network is typically represented by an "axial map", which can be constructed by drawing the least set of longest lines of uninterrupted permeability that are necessary to cover all the public space and make all the connections (Hillier & Hanson, 1984). The map can be automatically derived if we have an accurate representation of built forms (Peponis, Wineman, Bafna, Rashid, & Kim, 1998; Turner, Penn, & Hillier, 2005). The integration of a line in the axial map can be analyzed based

on its average distance from all the other lines in the system, with the distance defined as the smallest number of lines it needs to traverse to get to another line.

In addition to the axial map, the public open space can also be represented according to the standard conventions of street center-line maps typically used in GIS-based representations of street networks. Measures of connectivity and density computed based on street center-line maps have been proposed. For example, three measures of reach—metric, directional, and directional-metric—have been proposed by Peponis, Bafna, and Zhang (2008) to characterize street density, connectivity, and the associated urban potential. Metric reach measures the amount of street length accessible within a walking distance and is equivalent to the ideas of walk shed or walking catchment area (Hess, 1997; Hess, Moudon, Snyder, & Stanilov, 1999). Directional reach measures the density of street length accessible within a specified distance measured in turns, or direction changes. Directional-metric reach measures the amount of street length accessible within both a metric and directional distance limit. A parametric syntactic analysis is enabled by varying the distance limit and other parameters associated with those measures (Feng & Zhang, 2017, in press).

The main findings coming out of the field of space syntax related to urban research can be summarized in the following three aspects.

(1) The distribution of movement densities over an area is a function of the syntax of the street network

A large amount of empirical studies coming out of the field of space syntax have demonstrated that the varying degrees of integration have an independent and systematic

effect on the distribution of pedestrian and vehicular movement in the city (Hillier, Burdett, Peponis, & Penn, 1987; Hillier, Penn, Hanson, Grajewski, & Xu, 1993; Penn, Hillier, Banister, & Xu, 1998; Peponis, Hadjinikolaou, Livieratos, & Fatouros, 1989; Peponis, Ross, & Rashid, 1997; Read, 1999). The linear correlation between the connectivity of streets (measured by metric reach and directional distance) and the movement density holds even if land use and development densities are taken into account (Ozbil, Peponis, & Stone, 2011; Peponis et al., 2008).

These findings suggest that the street network modulates the patterns of co-presence, encounter and potential co-awareness that characterize urban space. They also imply that cites can be looked at as 'movement economies' (Hillier, 1996a). By regulating the distribution of movement, the street network also regulates the degree to which the interfaces between streets and buildings are exposed to movement, which, in turn, influences the location of certain land uses, such as retail. The presence of retail can further accelerate the effect of integration in attracting movement, as found in some case studies (Hillier et al., 1993).

(2) Distributed attraction according to the syntax of the street network

Recent research demonstrate that commercial activities not only concentrate around central places but also align themselves with better connected streets. Some of this work has linked measures of land use to measures of network centrality (Porta et al., 2009; Sevtsuk, 2010). Much relevant work has developed in the field of space syntax (Chiaradia, Hillier, Schwander, & Wedderburn, 2012; Hillier, 1997, 1999; Ortiz-Chao & Hillier, 2007). In the context of space syntax, the distribution of commercial land uses

has been linked to the distribution of movement. Better connected streets are more likely to be selected as destinations (based on closeness centrality) or to feature along paths to other destinations (based on betweenness centrality); by becoming more frequented, they support land uses that benefit from the presence of people that can become potential customers. Scoppa and Peponis (2015) have shown that the effect of street connectivity remains significant after controlling for the impact of other variables, including population and employment density, transit stations, distance from central places, and zoning.

The new models of the effect of street connectivity upon the distribution of land uses lead us to a new understanding of centrality. Centrality is not limited to a few central places that can be located as points on a map, but is potentially present everywhere and to varying degrees, according to the way in which the street network links local areas to their surroundings. Centrality, in other words, is the outcome of the dynamic interaction of places and networks, of parts and wholes.

(3) The resonance between spatial cognition and the syntax of street networks

While the relationship between spatial cognition and spatial layout has initially been studied in buildings (Haq, 2003; Haq & Zimring, 2003; Peponis, Zimring, & Choi, 1990), the link between cognitive maps of the city and the syntax of street networks is increasingly well understood as well. Better connected lines (as identified by integration analysis) feature more frequently in people's cognitive maps; more importantly, there is a homology between the actual patterns of connectivity and centrality of streets and the connectivity or centrality of streets as represented in cognitive maps (Kim & Penn, 2004).

By observing people's route choice decisions in a virtual test environment, Dalton (2003) pointed out that there are multiple competing logics in route selection, including the desire to select the least meandering route and the desire to maintain a heading closest to the destination from the origin.

The way in which people understand city-maps is also a function of the way in which they understand the actual city. In assessing centrality according to the examination of a street center-line map, people take into account not only metric network distance, but also the alignment of streets and the directional or angular distances involved (Christova et al., 2012; Sakellaridi et al., 2015). Based on space syntax, we can identify the portion of the street network which acts as the 'skeleton' of cognitive maps, a connected system of main streets to which we are able to link any number of specific locations, as we become increasingly familiar with a city (Kuipers, Tecuci, & Stankiewicz, 2003).

1.1.9 Deformed grid and space syntax

Intuitive observations

In space syntax, the term *deformed grid* has been used to describe the kind of arrangement of public open space frequently associated with historical towns and cities that have been developed with little centralized planning (Hillier, 1996b, 1999; Hillier et al., 1983). By showing Apt, a small town located in the south of France, as an example, Hillier et al. (1983) described the key characteristics of a deformed grid.

> What do we mean by a deformed grid? First, compared to an orthogonal grid, the length of sightlines from particular spaces—their one-dimensional extension—is

> sometimes restricted and sometimes extended…This one-dimensional extension we call axiality. Second, the width of spaces—their two-dimensional extension—varies considerably. This we call convexity. In Apt the buildings seem to be arranged in such a way as to create a continuous flow of open space with wider and narrower sections and shorter and longer perspectives. The 'streets' and wider spaces are always lined by entrances to buildings, leaving few areas of blank walls…Also the layout offers a choice of routes from any point in the town to any other, with few cul-de-sacs. (p. 50)

Hillier (1996b) also made a subtle point about the difference between an *interrupted grid* and a deformed grid. In an interrupted grid, streets either continue straight or turn ninety degrees, thus could also be called "zero-ninety grids". Therefore, in an interrupted grid, the change in the direction of movement is either zero degrees or ninety degrees. By contrast, in a deformed grid various angles of incidence of streets can be observed, therefore the change in direction of movement can be anything between zero and ninety degrees. He pointed out that "the commonest kind of grid is not interrupted but deformed" (p. 276).

Despite the subtle difference between an interrupted grid and a deformed grid, they are all *structured grids* in which the "lines and areas are prioritized for integration and intelligibility to varying degrees in order to create a system of differentiation". It is this differentiation that Hillier (1996b) called *structure* in a system (p. 269). The variation in the length of lines (as in axial maps) and the angles of incidence between lines are obvious means by which differentiation—hence structure—is created in an urban grid.

In addition, Hillier (1996b) argued that the deformed grids as exemplified in many urban settlements share nearly invariant configurational properties (or common deep structures) that are essentially products of two kinds of law—namely, "laws of spatial emergence, by which the larger-scale configurational properties of space followed

as a necessary consequence from different kinds of local physical intervention; and laws of 'generic function', by which constraints were placed on space by the most generic aspects of human activity, such as the simple facts of occupying space and moving between spaces" (p. 262).

Linking generative principles to global syntactic properties

Hillier (2002) argued that the social forces working through the spatial laws created both the differences and invariants in settlement forms. After examining axial maps of 58 cities originate from different cultures all over the world, he found that in spite of the differences in the geometry of the axial maps which seem to reflect distinct "spatial cultures", there are also powerful invariants that seem to go across cultures and even across scales of settlement.

The first striking similarity lies in the statistical distribution of line lengths—they were all made up of a very small number of long lines and a considerable amount of short lines, approximating a logarithmic distribution. By experimenting with different ways of placing objects in space and testing their effects on the gain of the mean universal distance for the whole spatial system, he showed that four different principles can be utilized to either maximize or minimize the "depth gain" in a system. In doing these experiments, he effectively established a link between generative principles and syntactic outcomes (as he also did in the chapter "Is architecture an ars combinatoria?" in Space is the Machine (Hillier, 1996b)). More specifically, he showed that the conservation of long lines at the expense of creating more shorter lines was a necessity for minimizing universal distance in the larger-scale system. In this regard, as the settlement grows, the

long straight streets seem to play a key role in creating an efficient and intelligent global structure.

The other invariant is that each settlement has an “integration core”, which is formed by the most integrated lines in the street system. Moreover, the integration cores almost invariably resemble a “deformed wheel”. The deformed-wheel-like integration core features “a hub, spokes in all main directions, sometimes a partial rim of major lines, with less integrated, usually more residential, areas in the interstice forms by the wheel” (Hillier, 2002, p. 159). He showed that, as a global pattern, the deformed wheel holds up remarkably well for both big cities and small towns.

According to Hillier (2002), there seem to be two processes operating in parallel in forming the global invariants and the local variants: “one a local process generating differences in local grid patterns and apparently reflecting differences in spatial culture in some way; and the other a global process generating a single overriding structure that seems to reflect a more generic or universal process of some kind” (p. 161).

Unlike Hillier who conducted theorical experiments to explicate the relationship between the local generative principles and the emergent syntactic properties, Serra et al. (2016) explored how incremental development leads to the emergence of global network properties through empirical evidence. In their study, they observed the incremental development which added new streets in the Oporto Metropolitan Area, Portugal over a time span of 60 years. The street layouts resulting from the individual pieces of development were sorted into different morphological types, or “typomorphologies”, with clustering techniques. Therefore, the typomorphologies were not preconceived and

imposed but rather a result of the quantitative analysis. Most interestingly, they found that there were significant correlations between the typomorphological ratios (i.e., the ratios of local street networks of different morphological types) and the global network properties evaluated based on common space syntax measures, such as integration.

Both their studies demonstrate the possibility of linking the generative principles applied at the local scale to the syntactic differentiations emerged at the global scale.

Differentiation in street grids: syntactic types

As pointed out by Figueiredo and Amorim (2007), in the field of space syntax, comparison between street networks is often made after first sorting them into classes defined by cultural or regional differences and then based on morphological terms (Asami, Kubat, & Istek, 2001; Karimi, 1997). By contrast, Figueiredo and Amorim (2007) proposed an emergent taxonomy for urban grids by applying an automatic classification method (more specifically, the "average linkage clustering") on a sample of city maps from 22 countries. They considered three variables, including the "aggregation degree", which indicates how "deformed" a grid is; the "descriptive improvement", which indicates the improvement in the correlation between line length and degree; and "grid coefficient", which indicates how much a street structure resembles a regular grid. Based on this method, they were able to find meaningful typologies of cities, such as the grid-like cities, the tree-like cities, and "deformed" cities which have medium values of the selected variables.

Peponis et al. (1989) distinguishes three types of urban layouts based on the relation of local layout patterns and overall global patterns: "continuous layouts have a

diffused pattern of centrality reaching toward all parts of the system"; "regionalized layouts are of two kinds: those whose parts converge towards a common center and those whose parts diverge towards distinct subcenters"; and "polarized layouts superimpose a large scale urban grid on a deformed small scale urban fabric in such a way that the two principles of organization cannot be geographically separated in a non-trivial way" (p. 43). He also proposed measures of how well an integration core reaches all the parts of a system (indicated by the measure of "spread") and how much the integration values of core spaces differ from others (indicated by the measure of "strength").

Peponis et al. (2015) examined a particular kind of street network where strong differentiation exists between the supergrids of primary roads and the inserted local streets. Four different syntactic principles were uncovered through theoretical experimentation, including concerns about whether the shortcuts within and between superblocks necessarily involve the supergrid, with the shortest distance defined by path length, direction changes, or the number of intersections to cross.

Feng and Peponis (2018) attempted a rigorous definition of "deformed grid" in superblocks with traversing local main streets by systematically generating, analyzing, and querying into a theoretical design space. One rigorous definition of the deformed grids was given: first, they have long trails; second, they have a high differentiation of directional distance values.

It is worth noting that the three studies reviewed above have also influenced our own study in specific ways. One of the measures we used to evaluate the regularity of a street network (the measure of which we called "fragmentality") is adapted from the

measure “aggregation degree” mentioned in Figueiredo and Amorim (2007). The concept of high/low clarity of an integration core we used to develop the typology based on the patterns of centrality is inspired by the measures of “spread” and “strength” introduced by Peponis et al. (1989). The overall research methodology developed in this study—which is built on three different kinds of algorithms (i.e., generative, analytical, and query algorithms)—is consistent with our previous study of the superblock designs (Feng & Peponis, 2018).

1.2 Research Program

1.2.1 Purpose and scope of research

The initial impetus for the study is the heuristic distinction between regular superblock grids, hierarchical branching patterns engendering discontinuous enclaves, and deformed grids comprising local main streets, as these are commonly found in many parts of the world. The purpose of this study is to offer formal definitions of different kinds of differentiation as they apply to the heuristic type of deformed grids, a type which is central to much work associated with space syntax literature[7].

This study builds on the intellectual foundation of space syntax and fits into the larger research themes that have been central to a syntactic interpretation of the urban geometry: (a) the investigation into the dynamic spatial processes in which sequences of

[7] However, in this study, we do not pursue the subtle difference between an interrupted grid and a deformed grid as articulated by Hillier (1996b). The basic definition of a deformed grid as mentioned earlier in the chapter is used instead.

local spatial moves give rise to specific spatial and syntactic patterns at various scales, (b) the study of the interplay between the geometric differentiation (e.g., the variations of line lengths and angles of incidence between lines) and the syntactic differentiation (e.g., the variation of integration) that gives structures to the urban grids, and (c) the articulation of emergent syntactic types expressed in the more complex structural biases identified as “integration cores”.

The focus of this research is thus a theoretical understanding of *differentiation*, treated as *variability of local conditions* and as *structure of centrality*. We use the term *differentiated grids* to describe street layouts in which the street segments are geometrically or syntactically differentiated. The kinds of differentiation discussed in this study are only engendered by varying the physical form of the street network. The streets themselves are treated as functionally uniform—in other words, we do not distinguish between paths or parts of street sections that are intended for specific categories of users such as pedestrians, cars, bicycles and so on. Besides, the differentiated grids are studied as planar systems. Therefore, we do not consider the pathways that run above or below ground level such as overpasses or underpasses.

1.2.2 Specific research questions

Specifically, we ask:

1. How to create differentiated grids by progressively deforming a square grid?
2. What different kinds of differentiated grids are there?
3. What is the relationship between the different rules that can be applied to creating differentiated grids and the emerging types of differentiation?

1.2.3 Research approach

We explore different kinds of differentiation in street grids at the scale of a superblock by systematically deforming a square grid that measures half a mile (800 m) on each side. A square grid bounded by a square is undifferentiated geometrically and syntactically. The reason we pick the square grid over the hexagonal grid or the triangular grid as the initial premise is twofold. First, while an infinite hexagonal grid or triangular grid also represents a geometrically and syntactically neutral state, it is bound to be differentiated along the edge when bounded by a square. Second, a square grid is much more often discussed and used in urban design and planning.

We have developed a total of eight different types of local, elementary, syntactic operations to deform a square grid of 9 × 9 streets in a stepwise manner. We also defined an additional type of operation, which we call "mixed operations", by applying selected types of operations in random sequences. These operations transform the underlying connectivity of the street grid, its geometric alignment, or both. For each type of operation, we further developed rules about where the operation can be applied, how it is applied, and, where appropriate, the parametric constraints that govern its application. A total of 5400 *theoretical designs* were thus generated by the application of these different types of operations, comprising a "design universe" for systematic analysis, comparison, and query.

1.2.4 Several limitations

First, no attempt is made to use the operations in order to generate street patterns that actually resemble particular empirical samples of interest. Nor do we demonstrate

whether the operations under consideration might be applied to improve particular street plans in specific ways—for example by addressing and optimizing configurational properties that have particular functional significance or implications as based on prior research. The study is limited to a theoretical exploration of local and global aspects of differentiation.

Second, no attempt is made to study the effects of applying specific subsets of operations in particular sequences, or according to rules intended to restrict the application of the operations to prespecified global ends—in this case particular kinds of centrality structures.

Third, no attempt is made to develop generative principles that operate at two or more levels: for example, by first generating a primary local grid and then proceeding to add infill streets.

1.3 Outline of Dissertation

The rest of the dissertation is organized into eleven chapters.

Chapter 2 introduces the individual operations we developed to deform a grid in a stepwise manner. In addition, it shows effects caused by a single application of each operation on a simple square grid.

Chapters 3–9 explain the generative rules developed to generate the differentiated grids. For each generative rule, we show how different measures vary as the corresponding operation is applied more frequently.

Chapter 10 analyzes the operations' and generative rules' capacities to affect individual measures. We also quantify the relationship between measures along multiple dimensions of differentiation, with linear regression models.

Chapter 11 introduces the typology of street networks we developed and algorithmically defined based on the size and shape of the integration cores. It also discusses the relationship between the generative rules and the emergent types of street networks.

Chapter 12 discusses the contribution of the thesis. It also extends it by applying the query algorithms developed in the previous chapter on some well-known conceptual schemes and real examples across the world. It also points out limitations of this study and directions for future work.

CHAPTER 2
SYNTACTIC OPERATORS: A STEPWISE APPROACH TO DEFORMING GRID-LIKE STREET NETWORKS

2.1 The Generation of Differentiated Grids as a One-Step Bottom-Up Process

Before we introduce the set of operations developed to deform street grids, we first, at the conceptual level, distinguish between a one-step process and a multistep process and between a top-down process and a bottom-up process in procedural modeling of street networks.

2.1.1 One-step process vs. multistep process

In a one-step process—no matter whether the street network is created by deforming an existing one or by incrementally growing from a single street segment—no intermediate state is explicitly recognized to serve as a renewed premise for the application of subsequent operations to form the complete street network. The whole process is driven by iterative applications of prespecified operations whose rules of application remain the same throughout. The final form of the street network thus shows the cumulative effect of the iterations. But, to use the vocabulary of Hillier and Hanson (1984), no "description retrieval" is embedded in the process.

By contrast, a multistep process is one in which intermediate states are explicitly recognized to serve as renewed premises for the modeling step that immediately follows. For example, the complete street network of a city may be generated in three steps according to the categories of streets that are intended for different scales of movement—for example, arterials, collectors, and local streets. The main streets generated in the

earlier steps would be fixed and not subjected to the generative rules applied in the later steps. A multistep process is often used to generate street structures that have clear hierarchies built in from the outset—therefore, the structure of differentiation, rather than emerges along the generative process, is largely given.

In the field of computer graphics, modeling the street pattern in a sequence of steps is a common practice, especially for the generation of large-scale to mid-scale urban street layouts (Chen, Esch, Wonka, Müller, & Zhang, 2008; Peng et al., 2016). Primary street network is generated first, then finer structures are generated in subsequent steps. The multistep process that has frequently been adopted in the generation of hypothetical urban forms indicates how we design the street networks hierarchically based on the functional classification of streets in practice (Figure 2).

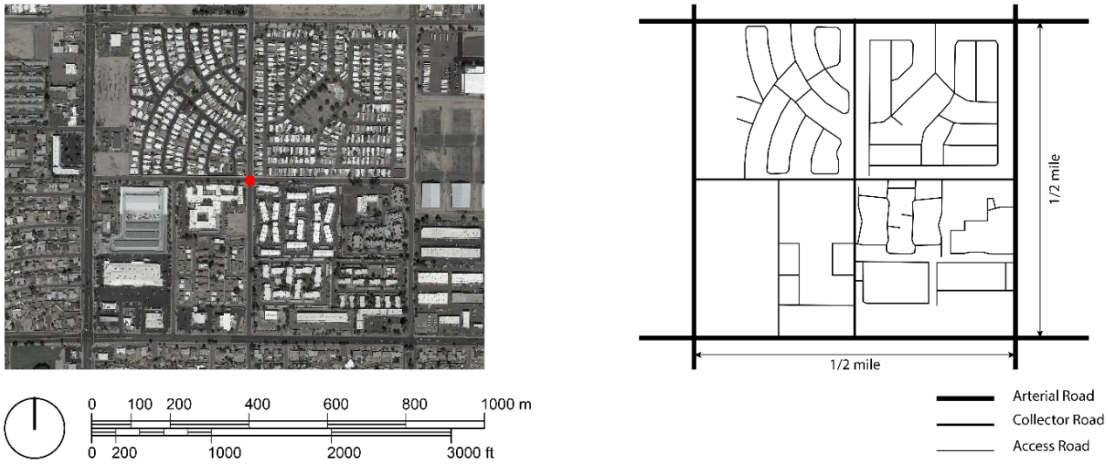

Figure 2. A superblock located in Phoenix, Arizona. The diagram on the right shows the hierarchical street structure inside the superblock. The satellite image on the left was created from Google Earth on April 28, 2017. The coordinates of the red dot in the image are 33°30'47.39"N and 112°10'54.95"W.

The parametric modeling exercise done by Peponis et al. (2016) in their study of superblocks in the Gangnam district of Seoul in Korea also follows such a process. As shown in Figure 3, the generation of the urban layout inside the perimeter blocks is predicated upon the placement of two traversing main streets. The pair of main streets divides the square area bounded by perimeter blocks into four quadrants which were further subdivided with distinct grid patterns. Because the main streets and the local street grids formed in both steps were imposed as a deliberate design choice rather than emerged from local operations, we may call this process more specifically as a two-step top-down process.

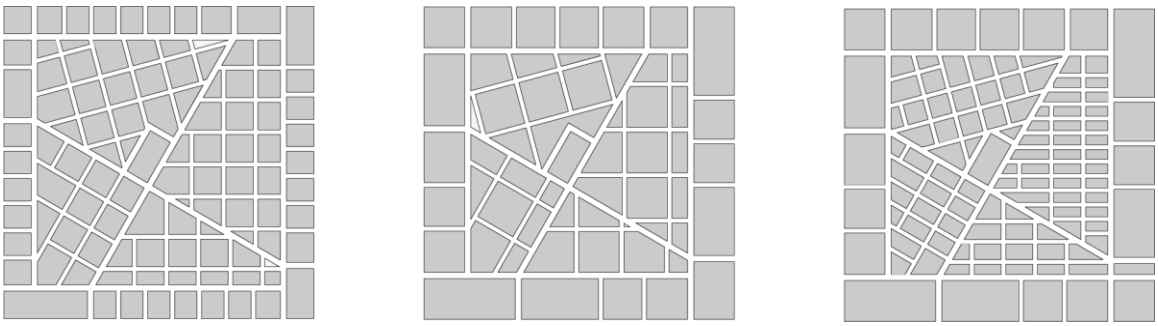

Figure 3. Hypothetical superblock designs sharing the spatial structure of Gangnam superblocks but encompassing a variety of blocks sizes.

2.1.2 Bottom-up process vs. top-down process

A top-down process is different from a bottom-up process in the exactness of control of the end results. A top-down generative modeling process leads to a preconceived pattern, while a bottom-up process leads to a pattern that cannot be fully predicted until the process is complete. Based on the distinction just described, it should not be difficult to imagine a two-step bottom-up process in which both the main streets

and the secondary streets are generated from the bottom up at each step. We can also combine bottom-up processes with top-down processes in a multistep process.

2.1.3 Why create differentiated grids by progressively deforming a square grid?

The generative process developed in this study is a one-step bottom-up process. We generated the differentiated grids by progressively deforming a square grid based on a set of individual operations that we collectively call *syntactic operators*. The syntactic operators are developed to vary the geometry and connectivity of streets at a local scale and can be repeatedly applied. We take this approach to generate and study the differentiated grids for several reasons.

First, it is a practical approach for exploring theoretical variations of deformed grids. Even at a limited scale, an exhaustive enumeration of deformed grids is simply impossible if we factor in both the composition (e.g., angle, length) and the configuration (e.g., connectivity) of streets. Even if we ignore the geometric variations of street grids and reduce it to a problem of graph enumeration, the number of variations soon becomes astronomical as the number of elements in the system increases to that normally found in a superblock.Second, it suits our interest in studying how the global structure of differentiation emerges from local variations in geometry and connectivity. Third, the square grid which has equal number of streets running in both directions is geometrically and syntactically undifferentiated. All streets inside it are equally connected—not only in terms of the number of streets that they are directly connected to, but also for the fact that for any street in the system, it takes no more than two turns to reach any other street in

the system. Hence it is a natural choice to use the square grid as a starting point to create differentiation.

2.2 Syntactic Operators

2.2.1 Local differentiation observed in historically evolved settlement forms

The initial inspiration for the syntactic operators developed in this study came from visual inspections of historically evolved urban settlements, in particular the French towns whose street forms very much resemble a "deformed grid" as described in space syntax literature (Hillier, 1996b; Hillier et al., 1983). In each of the French towns shown in Figure 4, there is a great variety of streets. They are long or short, sinuous or straight; yet, most of them are connected at both ends, forming blocks of varied shapes and sizes. Only a few cul-de-sacs are present—in each case, the street network is continuous and continuously differentiated.

There is also a variety of local patterns formed by intersection of streets (Figure 5). Four-way intersections and T-junctions are prevalent. Frequently we see two T-junctions placed close together and look as though they were simply produced by offsetting a continuous street in a four-way intersection—in fact, the opposite is true historically, since these proximate T-junctions are sometimes turned into cross intersections by street alignment aimed at rationalizing the plan. We treat the pattern made of the two T-junctions in the way just described as a unique local condition. (A few examples are highlighted by the yellow lines in Figure 5.)

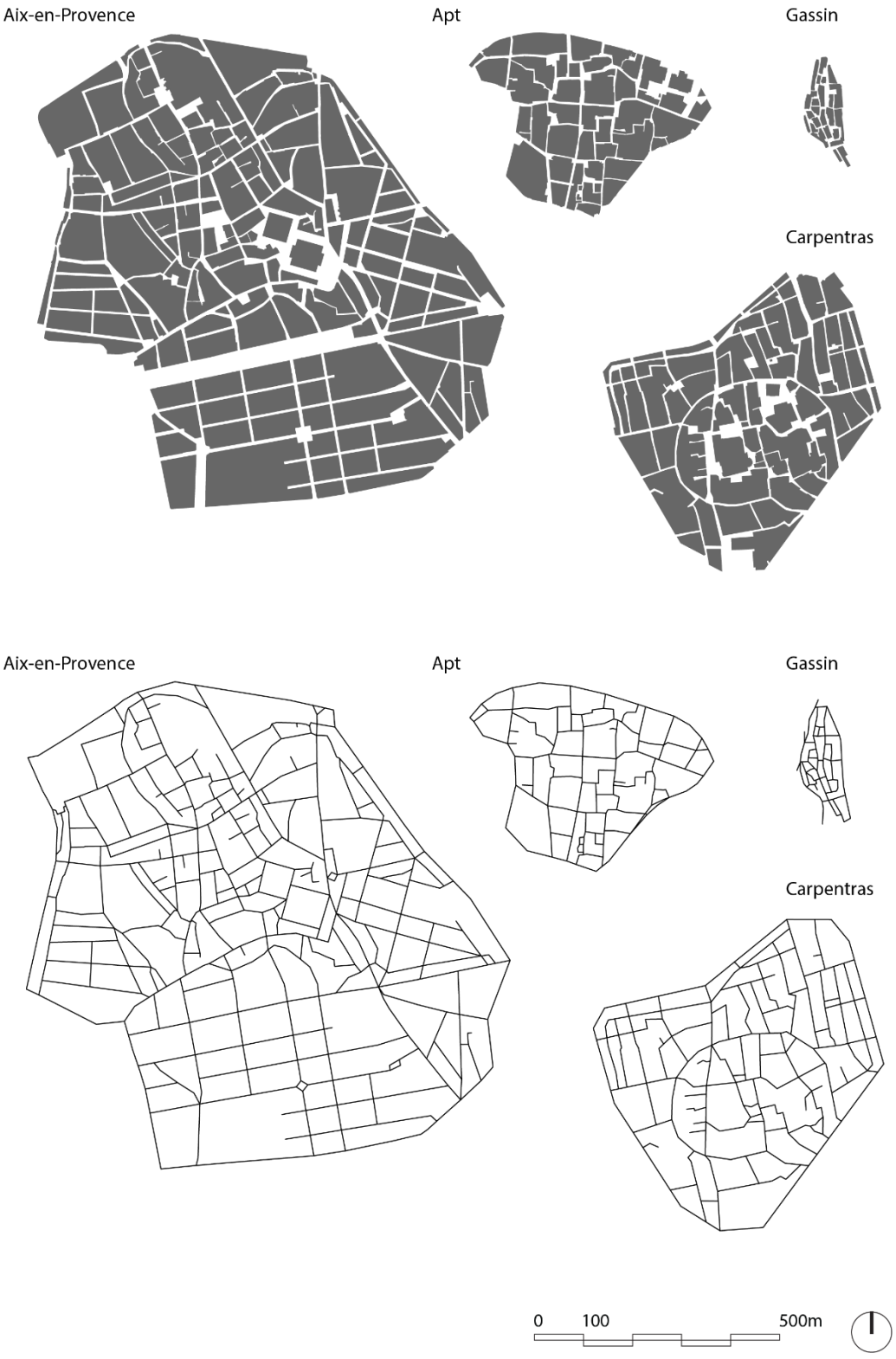

Figure 4. Maps showing the blocks and street centerlines of four French towns.

Occasionally we also see intersections composed of a greater number of street segments, or “arms” (highlighted by the red lines in Figure 5). For intersections that have the same number of arms, the geometric alignment of the arms can still differ—and this may be best seen in the case of four-way intersections. As two streets cross each other, one or both of them are bent to varying degrees as they pass the intersection. (A few examples are highlighted by the green lines in Figure 5.)

The intuitive observations of the distinct local conditions as defined by the individual patterns of street intersections became the initial source of inspiration for developing the syntactic operators. While our aim here is not to simulate the overall settlement forms in a particular sample of interest, we do aim to produce the different local conditions by applying the syntactic operators on a square grid—and preferably, each kind of local condition can be generated by applying one syntactic operator just once.

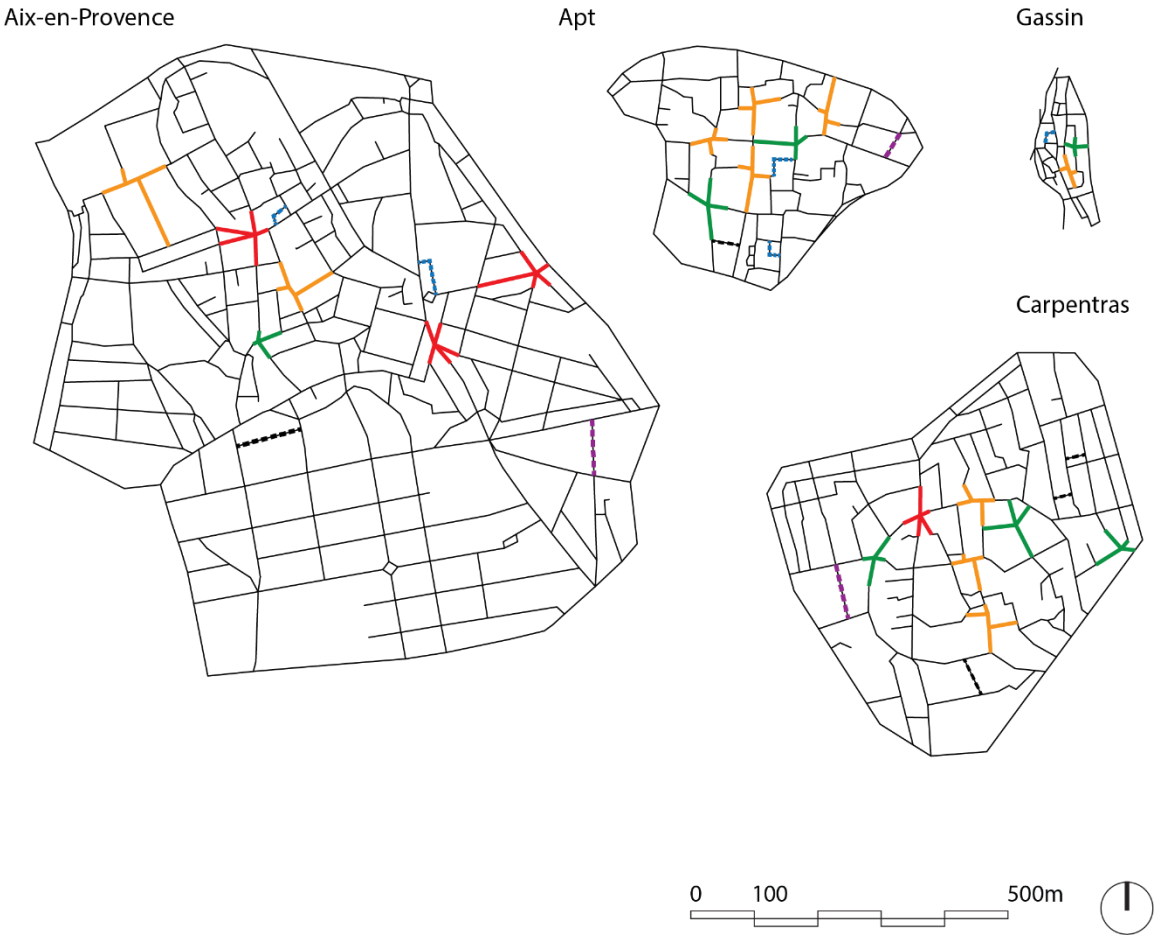

Figure 5. Illustration of different local conditions in four French towns.

2.2.2 Street graph: the basis for deformation

To deform a street grid, we first represent it as a *street graph*. In Figure 6, a 4 × 4 square grid[8] is represented by a street graph that consists of *vertices, edges, and cells*. In relation to the standard terminology of graph theory, every street graph being studied here is a *simple graph*—that is, each edge connects two different vertices and no two edges connect the same pair of vertices. They are *planar graphs* in that no edges can cross at any point other than their common endpoint. Each of them is also a *connected graph*—that is, there is at least one path between every pair of distinct vertices in the graph.

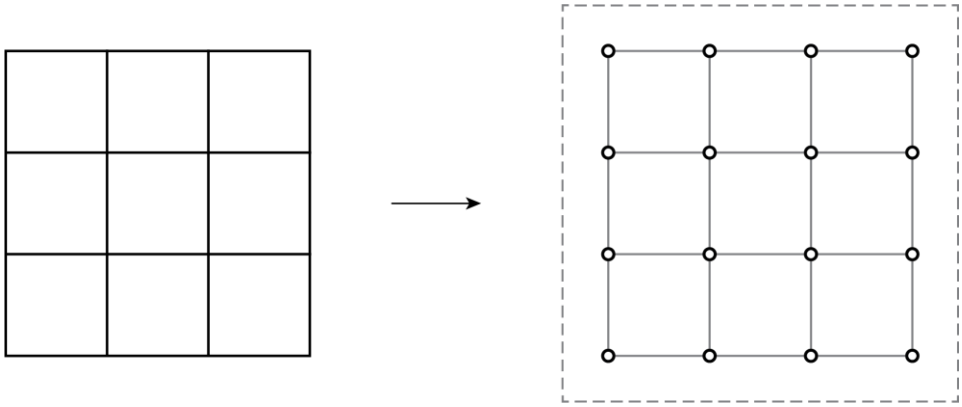

Figure 6. A 4 × 4 square grid and its graph representation. In the figure on the right, black circles represent vertices, solid lines represent edges.

[8] Here, a *m* × *n* grid indicates a uniform grid that has *m* streets running in the horizontal direction (or the *x*-axis) and *n* streets running in the vertical direction (or the *y*-axis). Since the main focus of study is not about the orientation of the street layout in relation to the four cardinal directions (i.e., north, east, south, and west), it is easier to refer to the horizontal and vertical axes as those in the Cartesian coordinate system.

The edges in the street graphs are drawn to indicate not only the pairwise relations between vertices but also the actual placement of street segments. Therefore, their intrinsic geometric attributes are as important as the adjacency relations that they represent. In this study, all edges are drawn as straight line segments. An edge is always associated with two vertices, while a vertex could be associated with one or more edges. A vertex is produced whenever street segments intersect. It also exists at the "dead end" of a cul-de-sac. Only vertices that have degree greater than or equal to three represent street intersections in the normal sense. What we call a cell is the polygonal area bounded by a continuous chain of edges, and it should not contain other cells. Since an edge is a line segment instead of a polyline, it can be associated with at most two cells. According to the definitions given above, a 4 × 4 grid has 16 vertices, 24 edges, and 9 cells. A data structure for recording the relationship between vertices, edges, and cells was developed to track changes in the original grid, which will be explained in more detail in the next chapter. Note that we have created an imaginary "wrapper cell" outside the boundary of the street graph so that the edges along the boundary are considered to be incident with the wrapper cell and can thus be analyzed in the same way as the internal edges (Figure 6). Except for the wrapper cell, a cell always represents an urban block.

2.2.3 Syntactic operators: local, elementary, and syntactic operations

The syntactic operators are operations performed on grid-like street graphs to produce new street graphs. In Figure 7, we use simple examples to illustrate the effect caused by each syntactic operator. When applied on a square grid—even just once—these operations can generate the distinct local conditions found in the French towns.

(1) Shift vertex

The operation of shifting vertex moves a vertex to a new position, and by doing so, it also moves the corresponding end of each incident edge to the new position. This operation has no effect on the total number of vertices, edges, and cells in the original street graph, but it changes the geometric alignment of streets. When applied on a uniform grid, this operation can vary the angle between the arms of a cross-intersection, thus creating local conditions similar to those found in the French towns (highlighted by the green lines in Figure 5). It has two clear effects: it changes the length of and the angular distance between the edges immediately affected—both of which affect local and global syntactic properties. From a designer's perspective, shifting vertex creates curvilinear streets and introduces variation in street vistas which is often wanting in a rigid grid-like urban layout.

(2) Contract edge

The operation of contracting edge presses the two endpoints of an edge together. The other edges originally incident with the two endpoints of the contracted edge may be varied geometrically because of the movement of one of their endpoints. When applied on a uniform grid, this operation always removes a vertex and an edge. It can create a vertex that have degree greater than four, thus creating radial street patterns similar to the local conditions found in the French towns (highlighted by the red lines in Figure 5). The syntactic properties are bound to change because of the change in the local pattern of connectivity, regardless of the induced variation in the length of and the angular distance

between the edges. From a designer's point of view, it breaks the monotony of grid plans and creates focal point of visual interest.

(3) Cross-concatenate vertices

The operation of cross-concatenating vertices joins two non-adjacent vertices that belong to the same cell (i.e. urban block). The edges incident with the two vertices may be varied geometrically because of the movement of one of their endpoints. Edges will be removed if they overlap existing edges and become redundant. This operation always removes one vertex from the original street graph. When applied on a uniform grid, it can create vertices that have degree greater than four, thus creating radial street patterns similar to the local conditions found in the French towns (highlighted by the red lines in Figure 5). The syntactic properties are bound to change because of the change in the local pattern of connectivity, regardless of the induced variation in the length of and the angular distance between the edges. From a designer's perspective, it breaks the monotony of grid plans and creates focal point of visual interest.

(4) Disjoin vertices

The operation of disjoining vertices removes an edge and thus "disjoins" the two endpoints of the edge. It should not be used for dangling edges (after deleting which there will be isolated vertices) or bridges (after deleting which the street graph becomes disconnected). After the operation, two cells merge into one. In consequence, while the total number of vertices will stay the same, the total number of edges and that of cells will be reduced by one, respectively. When applied on a uniform grid, it creates T-junctions that are ubiquitous in the French towns shown above. Because of the change in

the local pattern of connectivity, the syntactic properties are bound to change. From a designer's perspective, it reduces street density and removes connections between streets. However, it reflects the common practice of consolidating two or more urban blocks to accommodate special building programs such as hospital, stadium, or housing complex.

(5) Split vertex

The operation of splitting vertex splits an existing vertex in two and pulls them apart by "swinging" one of the incident edges along the edge that lies to the immediate left side of it (which would be a "left-split" with respect to the edge being swung) or to the immediate right side of it (which would be a "right-split" with respect to the edge being swung). This operation always increases the total number of vertices and edges by one, respectively. When applied on a uniform grid, it can create local conditions where two T-junctions are placed close together as often observed in the French towns (highlighted by the orange lines in Figure 5). Because of the abrupt change in the local pattern of connectivity, the syntactic properties are bound to change. From a designer's perspective, it discourages uninterrupted movement, and it is an effective way to allow but limit access to a space.

(6) Link vertex to vertex

The operation of linking vertex to vertex adds an edge to connect two existing vertices that belong to the same cell. It splits the affected cell into two and increases the total number of edges by one. When applied on a uniform grid, it can create vertices that have degree greater than four, representing street intersections that have more than four arms as observed in some of the French towns studied. By adding edges, it changes the

pattern of connectivity and further leads to changes in syntactic properties. From a designer's perspective, it creates diagonal streets, which are frequently used to optimize local circulation in grid cities.

(7) Link vertex to edge

The operation of linking vertex to edge adds an edge to connect an existing vertex and an existing edge that belong to the same cell. It splits the affected cell into two, increases the degree of an existing vertex, and creates a new vertex of degree three at the same time. Oftentimes, it amounts to the extension of a street at one end and the creation of a T-junction at the other end (so the street is not further extended), creating situations observed in the French towns (highlighted by the purple dashed lines in Figure 5). The syntactic properties are obviously influenced by the change in the local pattern of connectivity. From a designer's perspective, it can create local focus of visual interest and it introduces streets of different forms in a rigid grid city.

(8) Link edge to edge

The operation of linking edge to edge connects two existing edges that belong to the same cell with a new edge. In doing so, it creates two new vertices and splits the affected cell into two. It always creates T-junctions, which are common in the French towns studied. When applied on a uniform grid, it creates a short street perpendicular to two parallel streets on each side—a situation that we have also found in the French towns (highlighted by the black dashed lines in Figure 5). The syntactic properties change because of the added connection between streets. From a designer's perspective, it introduces short streets to optimized local circulation and creates a denser street network.

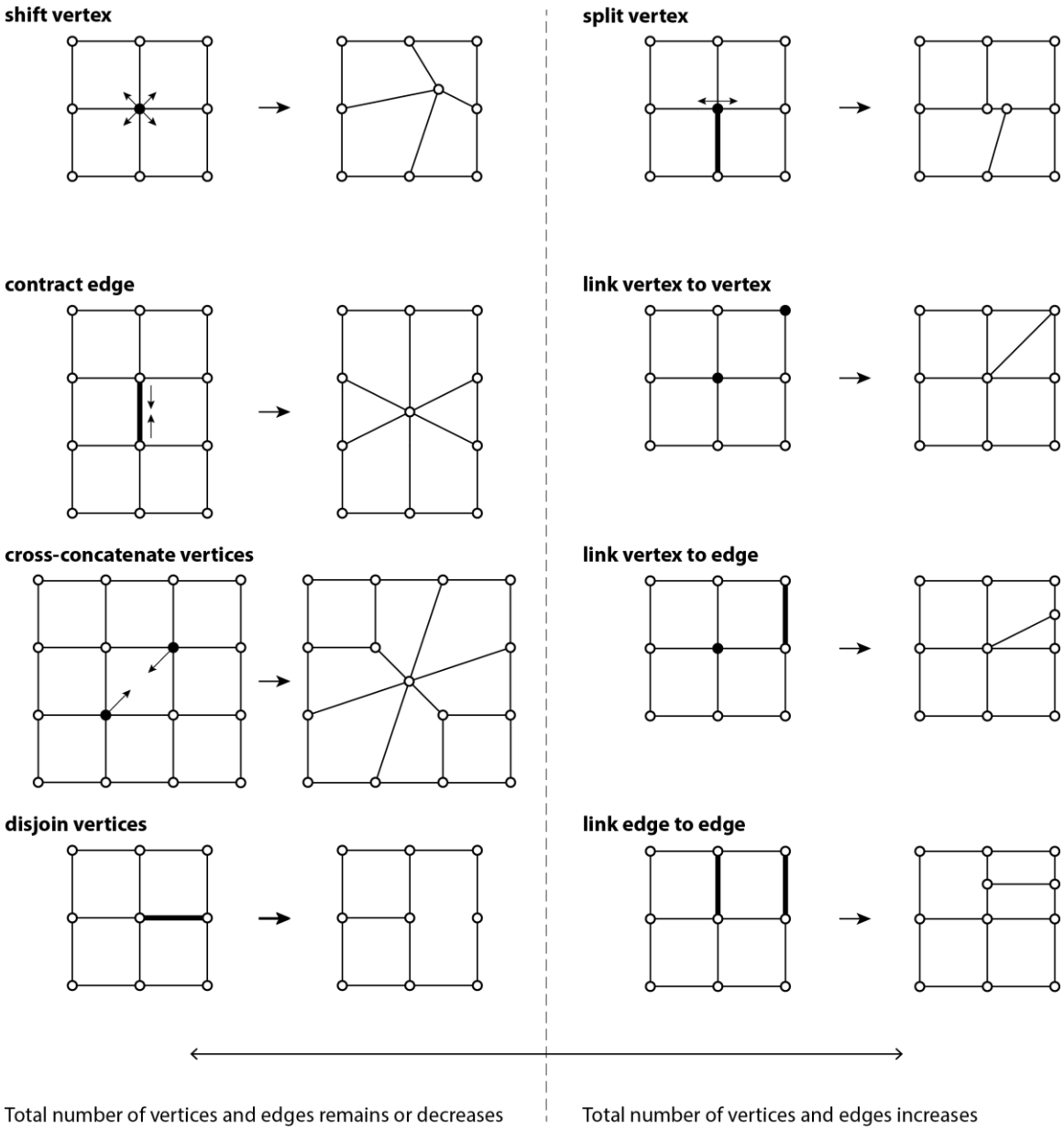

Figure 7. Syntactic operators applied on simple square grids.

Table 1 shows the potential changes in the total number of vertices, edges, and cells in *any* kind of street graph after each operation. For operations that increase the total number of vertices or edges, they have the potential to grow the original street graph into more elaborate systems. On the other hand, for operations that keep or decrease the total

number of vertices or edges, they have the potential to simplify or distort the original street graph, but they are incapable of growing the original graph. Figure 7 grouped the operations into left and right columns based on this distinction.

Table 1. Changes in the total number of vertices, edges, and cells after each operation

Name	Change in total number of vertices	Change in total number of edges	Change in total number of cells
Shift vertex	No change	No change	No change
Contract edge	- 1	- 1 or more	No change or decreased
Cross-concatenate vertices	- 1	No change or decreased	+1, no change, or decreased
Disjoin vertices	No change	- 1	- 1
Split vertex	+ 1	+ 1	No change
Link vertex to vertex	No change	+ 1	+ 1
Link vertex to edge	+ 1	+ 2	+ 1
Link edge to edge	+ 2	+ 3	+ 1

The syntactic operators are *local*, *elementary*, and *syntactic* operations. They are syntactic operations for the obvious reason that they all affect the syntactic properties of street networks (as networks of real space to be experienced by people, not as networks of links and nodes to be studied as pure mathematical structures). They are local in the sense that the elements subject to each operation in a street graph are confined to the components of one cell (equivalent to a single urban block)—if we consider the incident

edges and vertices that encircle a cell to be its components. These operators are elementary in the sense that they cannot be interpreted as repeated applications of another operator. They are elementary also in the sense that they do not have any "foresight" built in to achieve any predefined goals—hence they are best described as *local moves* instead of *local maneuver*.

Of course, except for linking vertex to vertex and disjoining vertices which involve no ambiguity when being performed, the other operations require some control of their behavior. A set of variables to parametrically control those operations have been developed and will be explained in greater detail in the following chapters.

2.2.4 How complete is this set of operators?

In terms of creating different local conditions on a bounded uniform grid by applying a *local*, *elementary* operation just once without (a) involving more than two vertices and (b) creating any cul-de-sacs[9], the syntactic operators introduced above represent an almost complete set of operations that one can think of. Below we explain why.

The pattern of street intersections can only vary in two ways: geometrically or topologically. Geometrically, we can vary the angle between the arms attached to the intersection; topologically, we can add or reduce the number of arms attached to the

[9] The idea of cul-de-sacs is inherently incompatible with the idea of a deformed gird as we described before.

intersection. Therefore, in the context of a street graph, the operations should be able to increase or decrease the degree of a vertex, as well as vary the geometric alignment of edges incident with a vertex.

The only way to vary a street intersection geometrically without changing its topology (i.e., the degree of the vertex at the intersection) is to move the corresponding vertex along with its incident edges—hence the operation of shifting vertex is included.

To change the topology of a street intersection, we can either increase or decrease the degree of the corresponding vertex. An obvious way to increase the degree of a vertex is to connect the vertex with other existing vertices—hence the operation of linking vertex to vertex is included. We can also increase the degree of the vertex by connecting it to a newly created vertex. However, the new vertex has to be located on an existing edge (otherwise it would produce a dangling edge and that amounts to creating a cul-de-sac which we would like to exclude)—hence the operation of linking vertex to edge is included. A perhaps less obvious yet very effective way to increase the degree of a vertex inside a uniform grid[10] is to concatenate it with an adjacent vertex—hence the operation of contracting edge is included.

On the other hand, the obvious way to decrease the degree of a vertex is to remove one of the edges incident to it—hence the operation of disjoining vertices is

[10] By "inside a uniform grid", we refer to the situations in the grid that resemble an infinite uniform grid. We treat the vertices located along the boundary of a uniform grid and their incident edges as special cases.

included. Another way to decrease the degree of a vertex inside a uniform grid is to swing one of the incident edges along another incident edge lies immediate to the left or right side of it—hence the operation of splitting vertex is included.

The operation of cross-concatenating vertices can increase and decrease the degree of existing vertices in a uniform grid at the same time, as shown Figure 7, hence it is also included.

We can, of course, create new local conditions by adding two new vertices. However, to meet the definition of a street graph (especially the condition of planarity) and not to create any cul-de-sacs, both vertices have to be located on existing edges—hence the operation of linking edge to edge is included.

So, does the set of syntactic operators introduced above include *all* the possible local operations dealing with one or two vertices? To answer this, we need to understand what all the possibilities are.

Table 2 shows all the theoretically possible operations that involve one or two vertices. Assuming that the two vertices subject to an operation always belong to the same cell, the validity of each operation listed there is tested against three initial premises of the syntactic operators: (a) it produces results that are still street graphs; (b) it cannot be interpreted as repeated applications of another operator; and (c) it creates distinct local conditions.

Table 2. Theoretically possible operations involving one or two vertices

Actions	One vertex	Two vertices
shift	Yes[a] (shift vertex)	Not valid[b]
add	Not valid	Not valid
delete	Not valid	Not valid
insert	Not valid	Not valid
insert and join	Yes (link vertex to edge)	Yes (link edge to edge)
join	Not valid	Yes (link vertex to vertex)
disjoin	Not valid	Yes (disjoin vertices)
concatenate	Not valid	Yes (contract edge; cross-concatenate vertices)
split	Yes (split vertex)	Not valid

Note.

[a] 'Yes' indicates that the corresponding operation is included in the set of syntactic operators. Inside the parenthesis are the names of the corresponding syntactic operators.

[b] 'Not valid' indicates that the theoretical operation is not a valid syntactic operator.

[c] 'No' would indicate that the corresponding operation is not included in the set of syntactic operators even though it would be valid.

Shifting two vertices is not a valid syntactic operator because the operation can be easily interpreted as applying the operation of shifting vertex twice. We distinguish between *adding* vertices and *inserting* vertices: to add a vertex means to place a new vertex anywhere except on an edge; to insert a vertex means to place a new vertex on an existing edge and split the edge into two. Adding vertices without adding edges to connect them to existing vertices will produce isolated vertices and make a disconnected graph (not a street graph anymore). Therefore, adding vertices—no matter one or more—

are not valid syntactic operators. Although inserting one or more vertices will produce valid street graphs, they do not create distinct local conditions (as discussed in the context of real space experienced by people instead of the mathematical space). Therefore, inserting vertices are not valid syntactic operators. To *delete* a vertex means to remove a vertex from a street graph without removing the edges incident to it. Since an edge always connects a pair of vertices in a street graph and deleting vertices will inevitably violate this condition, they are not valid syntactic operators. To *join* vertices means to connect two or more existing vertices with one or more edges. It involves at least two vertices. Therefore, joining a single vertex is not a valid operation. Likewise, disjoining a single vertex is not valid. To *concatenate* vertices means to combine vertices—be they adjacent or not. It always involves more than one vertex. Therefore, concatenating a single vertex is not valid. Splitting two vertices is not valid because it can be interpreted as applying the operation of splitting vertex twice.

It looks like the syntactic operators introduced above is indeed almost a complete set of *local*, *elementary*, *syntactic* operations that involve no more than two vertices and create no cul-de-sacs when applied on a bounded uniform grid just once. One may argue that there could be an operation about deleting a vertex along with all the edges incident to it. However, the effect it causes when applied on a uniform grid is a bit too dramatic—a humongous block of the size of four normal blocks without any streets inside, which is more like a construction site than anything we would find in a historically evolved city. Admittedly, one may also split a vertex differently. An alternative way to split a vertex is shown in Figure 8, the two vertices that result from the split are not connected by any edges. This, arguably, has the advantage to create a trivial loop formed by a short street

with an abrupt right turn in the middle which we did find in the French towns (as highlighted by blue dashed lines in Figure 5). However, in the French towns we studied, the trivial loops rarely occur in pairs as though they are part of a larger block as shown here, whereas the way we split a vertex can create the local conditions that are much more common in those towns, as described before.

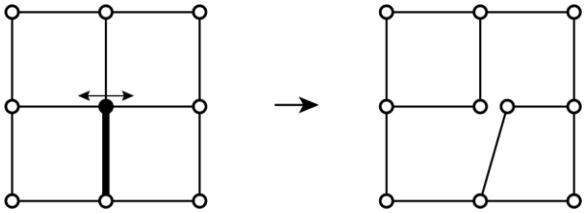

Figure 8. An alternative way to split vertex.

2.2.5 Intermediate street graphs vs. final street graphs

An *intermediate street graph* is a street graph that has just undergone an operation or a sequence of operations and may have collinear edges in it. A *final street graph*, on the other hand, is always converted from an intermediate street graph by amalgamating collinear edges at the end of the generative process. The difference between the intermediate street graph and the final street graph is only defined in the mathematical space instead of the real space as experienced by people[11]. In terms of the number of vertices and edges used, the final street graph is the most economical representation of the *same* street pattern delineated by the intermediate street graph. The final street graph

[11] For this reason, the operation used to convert an intermediate street graph to a final street graph is not a valid syntactic operator.

can be generated based on the intermediate street graph via a *polishing process* during which the redundant vertices are removed and the collinear edges are amalgamated. The polishing process is only initiated at the end of a generative process because it may exclude certain possibilities if applied early in the process. For example, in Figure 9, there is an additional vertex inside the square boundary of the intermediate street graph shown on the left, therefore in principle it is possible to create a cul-de-sac by disjoining vertices at a later stage. However, the final street graph shown on the right is completely devoid of that possibility. Similarly, it is possible to produce a 3 × 3 grid based on the intermediate street graph by applying operations of linking vertex to vertex, but it is impossible to do so by linking vertices in the final street graph.

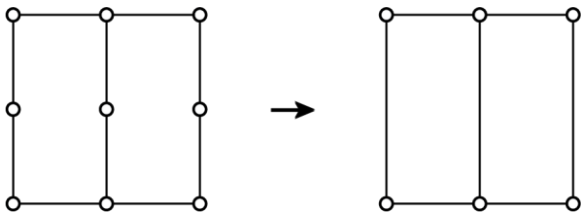

Figure 9. A simple example showing the polishing process.

2.2.6 Looking at syntactic operators from other perspectives

Operation vs. rule

We make a distinction between operations and rules. Operations are general descriptions of how to deform a gird-like street graph at a single step, while rules specify how exactly the operation is performed, when and where the operation should be performed, and when to stop.

Are they Euler operators?

The syntactic operators introduced in this chapter may remind readers of Euler operators used in solid modeling (Eastman & Waiter, 1979; Mantyla, 1984). However, the syntactic operators originated on different grounds. Certain syntactic operators also differ from Euler operators in the very nature of their behaviors.

Initially, Euler operators were introduced to the field of computer graphics to simplify the description of boundary representation which is a popular approach to model solids. They are a small set of data structure constructors which can help construct meaningful boundary representations in a stepwise and errorproof fashion. Euler operators got their name from the Euler-Poincaré formula. For a solid whose surface is a 2-manifold, suppose it has *s* shells and their face decompositions include *f* faces, *e* edges, *v* vertices, and *h* holes, then the Euler-Poincaré formula describes their relationship as follows:

$$v - e + f = 2 * (s - h) \qquad (1)$$

Euler operators add or remove vertices, edges, faces, holes, and shells in a manner that ensures the above equation always hold true. If we interpret the above equation as a four-dimensional hyperplane in the five-dimensional lattice with axes <*v, e, f, h, s*>, then Euler operators act as a set of transition vectors which span the four-dimensional plane (Braid, Hillyard, & Stroud, 1980). In other words, they are sufficient to generate every <*v, e, f, h, s*> tuple. Mantyla (1984) proved that "Euler operators are sound and complete in the family of objects bounded by physical 2-manifolds in the sense that a meaningful

geometry can be assigned to each model constructed by them, and that each 2-manifold can be modeled by them" (p. 56).

In addition to the trivial differences between Euler operators and the syntactic operators—such as that Euler operators have been used in 3-D solid modeling, while the syntactic operators as presented here were conceived only for manipulating 2-D geometries—there are several fundamental differences between them:

(1) While Euler operators speak to the abstract computer data structure involved in solid modeling, the syntactic operators speak to concrete and meaningful design moves that effectively deforms a street network. Euler operators, when interpreted as transition vectors, need not be a fixed set of operators. They can be varied to suit specific needs so long as they span the four-dimensional hyperplane determined by the Euler-Poincaré formula and ensure that the overall topological property is preserved. By contrast, specific design effects were considered in developing each of the syntactic operators, making them more than pure mathematical transformations.

(2) While Euler operators always result in changes to the topology of the object, the syntactic operators can result in changes only to the geometry with its topology intact (as the operation of shifting vertex does)[12]. This is because Euler operators

[12] The syntactic operators are not based on theoretical graph transformations where the changes in the geometric alignment of edges are often not a concern. In that sense, they are very different from the actions and rules used to transform graphs studied by Dalton and Kirsan (2008).

themselves do not need to address the geometric attributes explicitly: a geometric/physical realization can be acquired later, corresponding to the underlying topology.

(3) While the impact of each Euler operator on the topology of an object is predetermined and hence entirely predictable, the impact of some of the syntactic operators on an object's topology is not entirely predictable. Unlike Euler operators, proper topology is implicitly maintained during the syntactic operations rather than explicitly built into their definitions.

With the above differences in mind, it is not surprising to find that applications of the same Euler operator can result in dramatically different consequences from a designer's point of view, depending on the specific protocol used for the geometric realization of such operations. For example, Figure 10 shows two applications of the same Euler operator known as "make edge and vertex" (which is often denoted as "MEV"). They have the same effect on the number of vertices, edges, and faces in the street graph: both add one new vertex and one new edge to the original graph. However, they have led to two dramatically different designs.

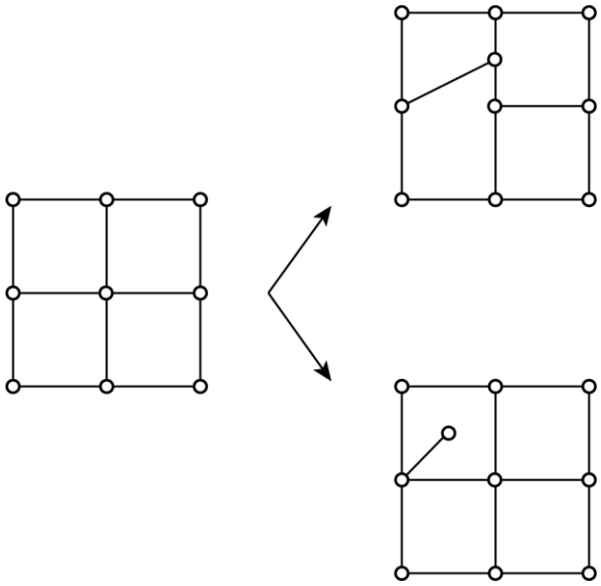

Figure 10. Applications of the same Euler operator can result in dramatically different consequences from a designer's point of view.

2.2.7 A brief note on the modeling space of syntactic operators

The street graphs to which we apply the syntactic operators are finite, simple, connected, planar graphs. Suppose a street graph has *v* vertices, *e* edges, and *f* faces (or *cells* as referred to in this dissertation), Euler's formula states that:

$$v - e + f = 2 \tag{2}$$

The minimal street graph has just a single vertex and a face (i.e. the outer, infinitely large region), thus satisfying the above equation. To ensure Euler's formula hold true, we need to make sure that the increase in the total number of vertices and faces equals the decrease in the number of edges in the modified street graph, and vice versa. We can denote this requirement as

$$\Delta v - \Delta e + \Delta f = 0 \quad (3)$$

If we check this requirement against the definite changes in the total number of vertices, edges, and cells caused by the syntactic operations (Table 1), we should find that this relation always holds true. Note that if we interpret $(\Delta v, \Delta e, \Delta f)$ as a transition vector, we only need two linearly independent transition vectors to span the two-dimensional hyperplane determined by Equation 3. As shown in Table 1, the syntactic operators, when interpreted as pure transition vectors, are sufficient to generate every $(\Delta v, \Delta e, \Delta f)$ tuple that satisfies Equation 3. In other words, by sequentially applying one or more of the syntactic operations, we should be able to generate any valid combination of *v*, *e*, *f* ("valid" meaning it can be mapped to a proper street graph), regardless of its specific geometric realizations.

2.3 The Syntactic Effect Caused by A Single Application of Each Operator

In this section, we study the syntactic effect caused by a single application of each operation on a simple bounded square grid based on two measures: the average vertex degree and the mean directional distance per length (DDL) for the street graph as a whole. The degree of a vertex *v* is the number of edges incident with it and denoted by *deg(v)*. The average vertex degree of the street graph is then simply calculated by summing up the degrees of all vertices and then dividing it by the total number of vertices in the street graph.

Peponis et al. (2008) laid the theoretical foundation of metric reach, directional reach, and directional distance. One way to calculate the mean DDL of a street graph is to

sum up the DDL values for all the edges (i.e., line segments) in the street graph and then divide it by the total number of edges. It can be simply denoted as

$$D = \frac{1}{|E|}\sum D_i \tag{4}$$

where D_i is the DDL value for the edge e_i. Although the calculation is simple and straightforward, it is dependent on the total number of edges which is again subject to the way in which we draw the street graph. Therefore, after being "polished", since the total number of edges may be reduced, the mean DDL of the final street graph calculated in this way may be different from that of the intermediate graph from which it is converted, despite the fact that the two street graphs represent the very *same* street pattern. Even if we agree that the calculation should always be based on the final street graph (i.e., the most economical representation) and hence rule out the discrepancy, it is still questionable whether we should weigh each edge equivalently in computing the mean value regardless of the respective length. If we agree that a person can potentially occupy any point along an edge or a street segment, and the overall experience of the street network is attained not only by looking through but also moving through each individual street, then the length of the street segment or the edge obviously matters.

For the above reason, in this study, unless otherwise specified, the mean DDL of a street graph is calculated as follows:

$$D_L = \frac{1}{L}\sum D_i l_i \tag{5}$$

where D_i is the DDL value for the edge e_i, l_i is the length of e_i, and L is the total edge length (i.e., the total street length) in the street graph. The mean DDL for the street graph is here denoted by D_L to emphasize that it is length-weighted. The mean DDL calculated as such will stay the same after the intermediate street graph is converted to the final street graph.

The directional distance or turning distance between two edges can be parametrically defined based on the angle of deviation involved in transitioning from one edge to another edge. As Figure 11 shows, if we register a direction change whenever the deviation angle is greater than zero (i.e., whenever the two edges are noncollinear), then splitting a vertex always adds at least one additional turn between the edge *a* and the edge *b*. However, if we register a direction change only when the deviation angle is greater than the angle β (β < α) as shown in Figure 11, then the turning distance between the two edges, after the operation, will increase by two in the situation shown in Figure 11a but will stay the same in the situation shown in Figure 11b. If we take any angle of deviation greater than *x* degrees as a change of direction, then the DDL calculated based on this assumption is called *DDL with x-degree threshold*, and simply denoted as DDL*x*d. For instance, DDL with 20° threshold is simply denoted as DDL20d.

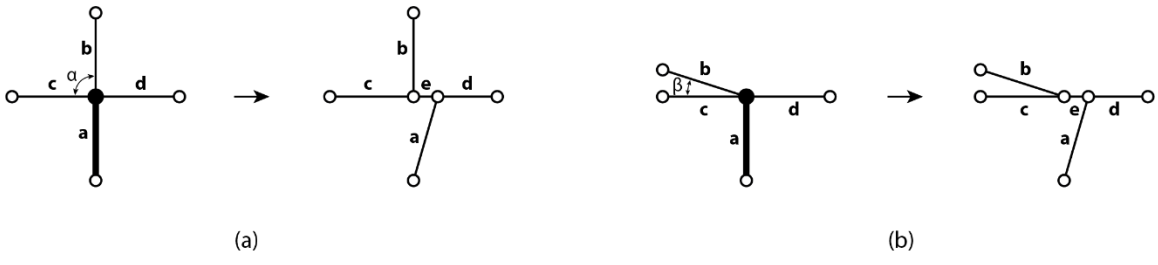

Figure 11. Two cases of splitting a vertex. The added turning distance can be parametrically defined.

In the rest of this section, we show the syntactic effect of each operation based on a perfect 4 × 4 square grid. For each operation, we enumerated all the non-equivalent outcomes. They are not equivalent in the sense that they cannot be transformed from one to another based on the symmetry group of the square. Then for each outcome—including both the intermediate street graph and the final street graph—we measured the average vertex degree and the mean directional distance per length (DDL) with 20° threshold, which are two important indicators of syntactic properties of a street network. Both the intermediate and the final street graphs are included in the comparison. As a yardstick to compare with, the average vertex degree of the original 4 × 4 square-grid is 3, and its mean DDL20d is 1.25.

As mentioned, for some operations, we had to specify certain geometric constraints to arrive at a definitive set of deformed street graphs. Therefore, it is more appropriate to call the operations studied here *restricted operations*.

2.3.1 Shift vertex

As Figure 12 shows, if we keep fixed the boundary[13] of the initial grid and only shift the vertex to the midpoint of one of the edges that are incident with it, then we can produce two non-equivalent street graphs by applying this operation just once on a 4 × 4

[13] By "keeping the boundary fixed", we mean that the number and position of the vertices and edges that comprise the boundary of the grid should not be changed during the generative process. Vertices on the boundary may only be removed by polishing at the end of the generative process when we produce the final street graphs.

grid. A consistent effect can be seen here: the mean DDL20d increased, and the average vertex degree stays the same because the topology of the street graph did not change.

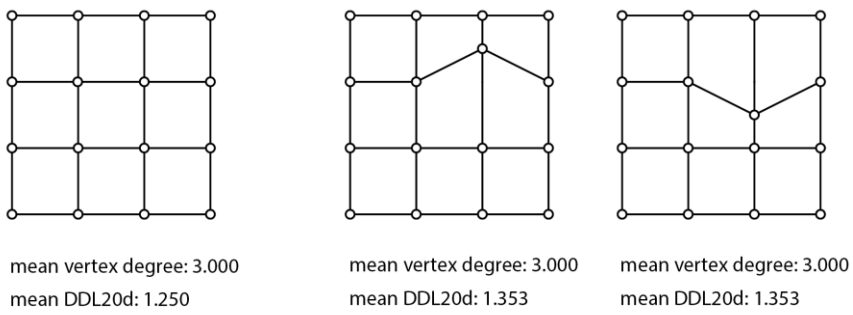

Figure 12. Non-equivalent street graphs yielded by applying the operation of shifting vertex exactly once on a 4 × 4 grid. Here, the final street graphs are the same as the intermediate ones.

However, note that we can parametrically define what counts as a direction change. If we make the threshold of the deviation angle sufficiently large, the mean DDL can be largely maintained as before.

2.3.2 Contract edge

As Figure 13 shows, if we keep fixed the boundary of the initial grid and contract the edge to its midpoint, then we can only produce one unique street graph by applying this operation exactly once on a 4 × 4 grid. After the operation, both the average vertex degree and the mean DDL20d are increased.

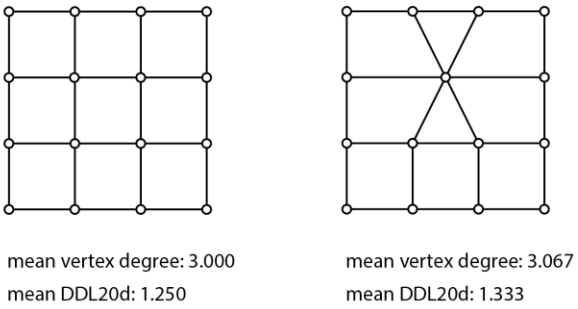

Figure 13. Non-equivalent street graphs yielded by applying the operation of contracting edge exactly once on a 4 × 4 grid. The final street graphs are the same as the intermediate ones.

Note that this is the only operation among all operations being studied that—when applied without any restriction—can reduce any street graph to a single vertex. Figure 14 illustrates a sequence of edge contractions after which the original street graph is reduced to a single vertex.

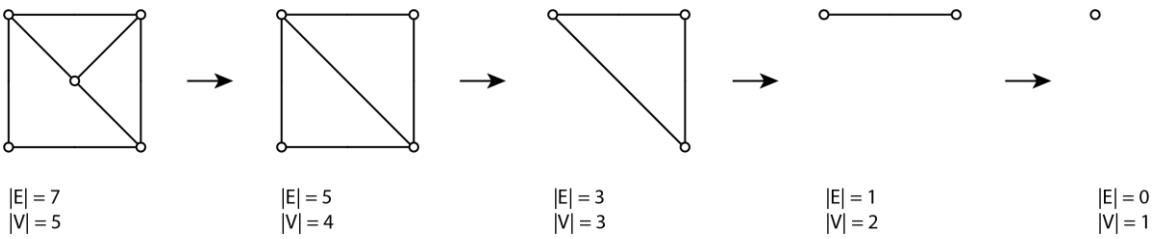

Figure 14. Repeatedly contracting edges eventually reduces a street graph to a single vertex.

2.3.3 Cross-concatenate vertices

As Figure 15 shows, if we keep fixed the boundary of the initial grid and cross-concatenate two vertices so that they become a single vertex located midway between their original positions, then we can only produce one unique street graph by applying

this operation just once on a 4 × 4 grid. After the operation, the average vertex degree is reduced and the mean DDL20d increased.

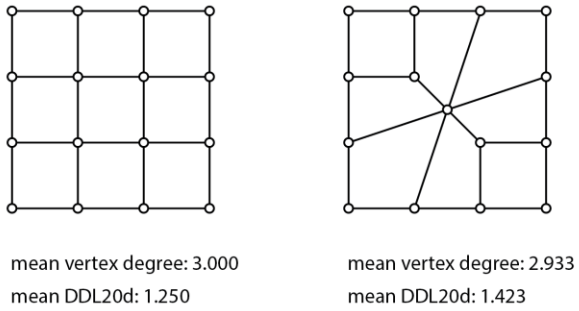

Figure 15. Non-equivalent street graphs yielded by applying the operation of cross-concatenating vertices exactly once on a 4 × 4 grid. The final street graphs are the same as the intermediate ones.

2.3.4 Disjoin vertices

As Figure 16 shows, if we keep the boundary of the initial grid fixed, then we can produce two non-equivalent intermediate street graphs by applying the operation of disjoining vertices just once on a 4 × 4 grid. A consistent effect can be seen in both cases: the average vertex degree is reduced and the mean DDL20d increased. In fact, according to the handshaking theorem[14], this operation always reduces the average vertex degree of the intermediate street graph.

[14] The handshaking theorem: Let G = (V, E) be an undirected graph with *m* edges. Then $2m = \sum_{v \in V} \deg(v)$. (Rosen, 2012, p. 653)

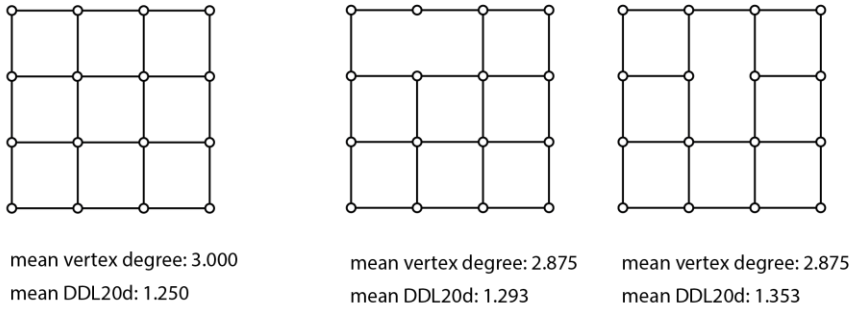

Figure 16. Non-equivalent intermediate street graphs yielded by applying the operation of disjoining vertices exactly once on a 4 × 4 grid.

As shown in Figure 17, a similar trend can be observed based on the final street graphs as well.

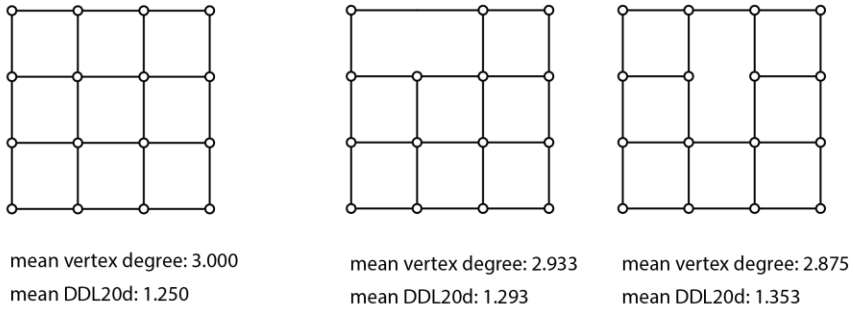

Figure 17. Non-equivalent final street graphs yielded by applying the operation of disjoining vertices exactly once on a 4 × 4 grid.

2.3.5 Split vertex

As Figure 18 shows, if we keep the boundary of the initial grid fixed and restrict the position of the newly split vertex to the midpoint of the edge along which it slid, then in total we can produce four non-equivalent street graphs by applying this operation on a

4 × 4 grid only once. A consistent effect can be seen: the average vertex degree is slightly reduced and the mean DDL20d increased. Besides, the total number of street intersections is increased by one because a cross-intersection is transformed into two T-junctions.

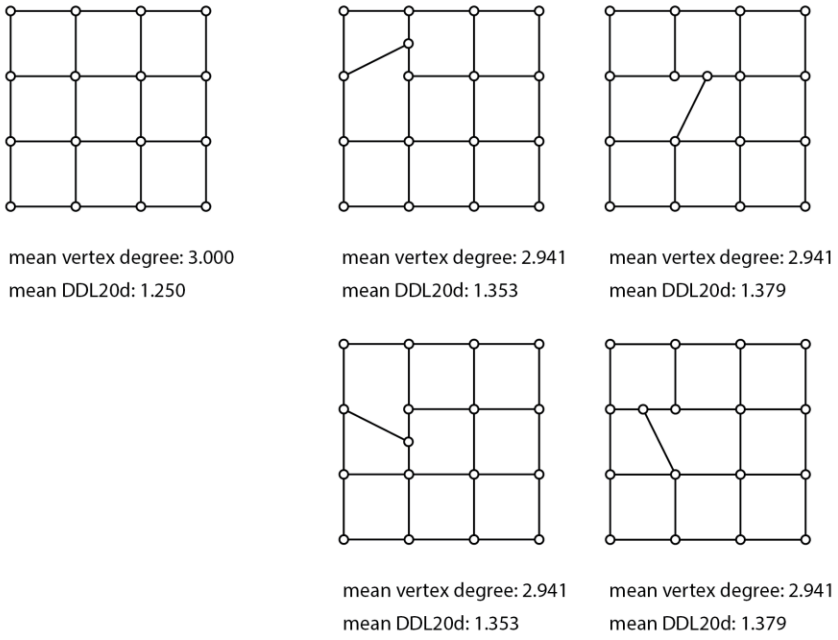

Figure 18. Non-equivalent street graphs yielded by applying the operation of splitting vertex exactly once on a 4 × 4 grid. Here, the final street graphs are the same as the intermediate ones.

It is not hard to prove that splitting vertex will always maintain or reduce the average vertex degree for a street graph that has at least one cell other than the wrapper cell. Based on the handshaking theorem, the average vertex degree of the original street graph can be calculated by the formula

$$\overline{deg} = \frac{2|E|}{|V|} \tag{6}$$

where $|E|$ indicates the total number of edges and $|V|$ indicates the total number of vertices. Similarly, based on the handshaking theorem and Table 1, we know that the average vertex degree of the street graph after the operation is

$$\overline{deg'} = \frac{2(|E| + 1)}{|V| + 1} \tag{7}$$

The change in the average vertex degree of the street graph can be evaluated as follows:

$$\frac{\overline{deg'}}{\overline{deg}} = \frac{2(|E| + 1)}{|V| + 1} \times \frac{|V|}{2|E|} = \frac{|E||V| + |V|}{|E||V| + |E|} \tag{8}$$

Now it all boils down to the comparison between $|V|$ and $|E|$: if $|V| > |E|$, then the average vertex degree is increased; if $|V| < |E|$, then it is decreased; if $|V| = |E|$, then there is no change. For a street graph that has at least one cell other than the wrapper cell, we have $|V| \leq |E|$ and the average vertex degree is at least two. Therefore, the operation of splitting vertex always maintains or decreases the average vertex degree of the original street graph.

2.3.6 Link vertex to vertex

Figure 19 shows the non-equivalent intermediate and final street graphs that can be produced by applying the operation of linking vertices on a 4 × 4 grid just once. A consistent effect can be seen: both the average vertex degree and the mean DDL20d are slightly increased. In fact, the average vertex degree will always increase after this

operation because the newly added edge will increase the total vertex degree by two while the total number of vertices will stay the same.

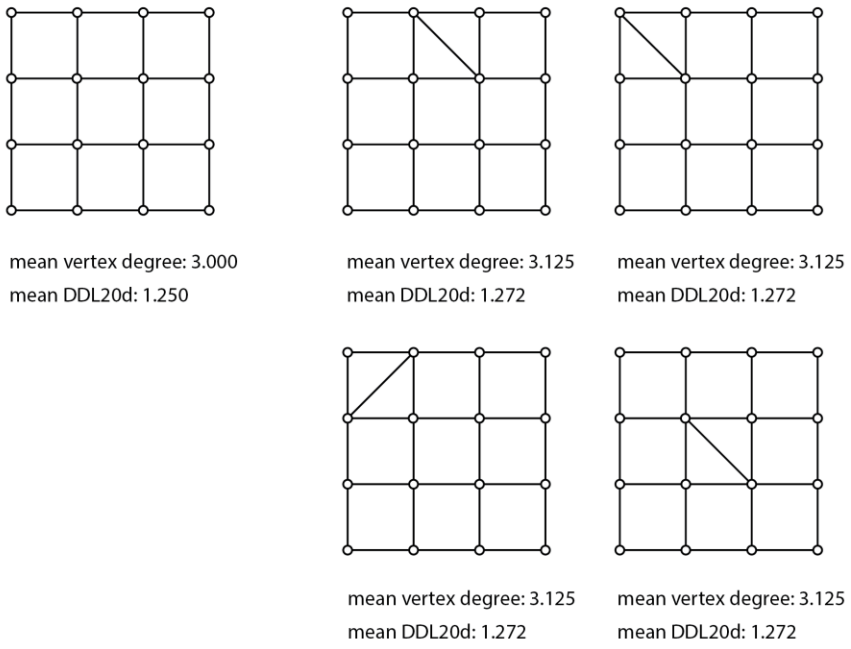

Figure 19. Non-equivalent street graphs yielded by applying the operation of linking vertices exactly once on a 4 × 4 grid. Here, the final street graphs are the same as the intermediate ones.

2.3.7 Link vertex to edge

As Figure 20 shows, if we are only allowed to link a vertex to the midpoint of an edge of its incident cells, then in total we can produce nine non-equivalent street graphs by applying this operation on a 4 × 4 grid just once. A consistent effect can be seen across all cases: both the average vertex degree and the mean DDL20d are slightly increased.

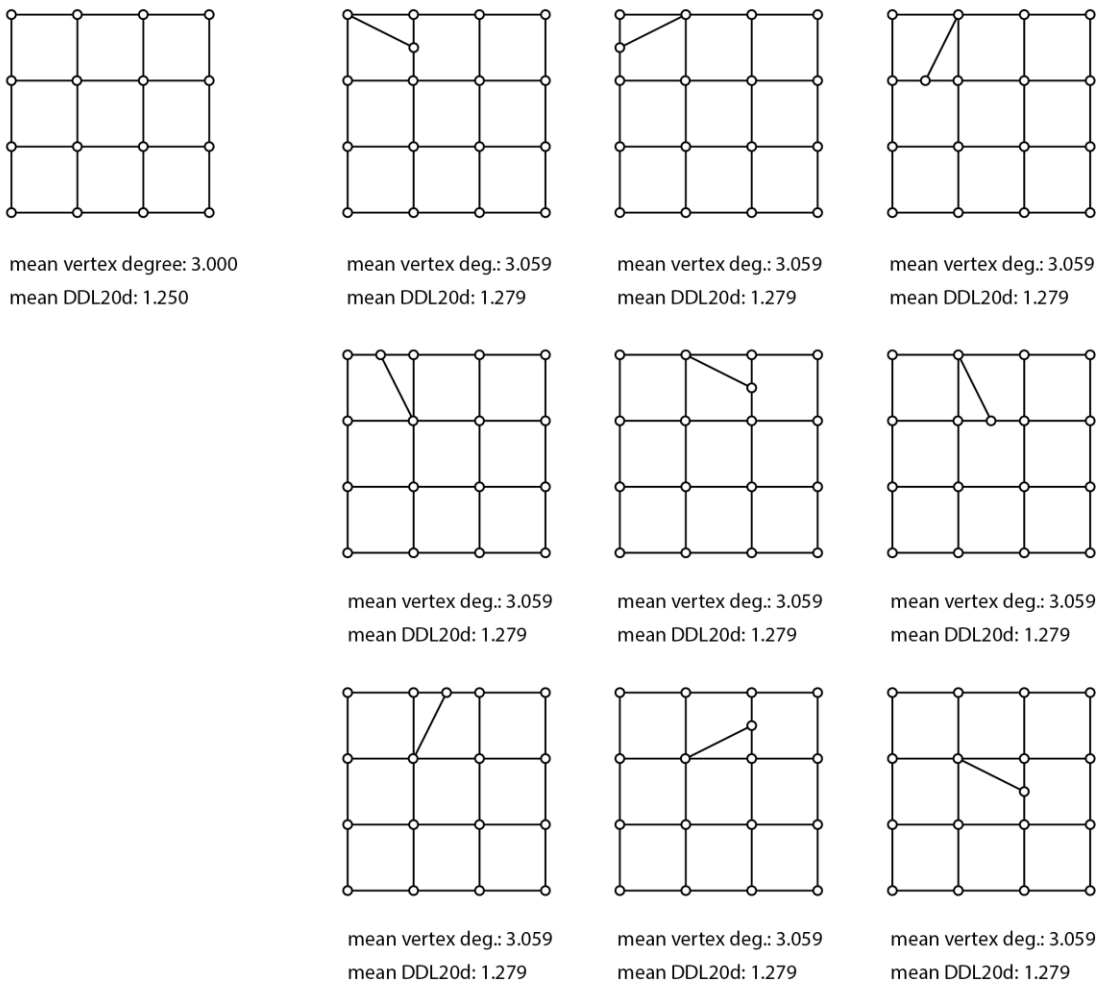

Figure 20. Non-equivalent street graphs yielded by applying the operation of linking a vertex to an edge exactly once on a 4 × 4 grid. Here, the final street graphs are the same as the intermediate ones.

2.3.8 Link edge to edge

As Figure 21 shows, if we are only allowed to link the midpoints of two opposite edges, then in total we can produce four non-equivalent street graphs by applying the operation of linking edges on a 4 × 4 grid just once. A consistent effect can be seen: the average vertex degree stays the same and the mean DDL20d is increased.

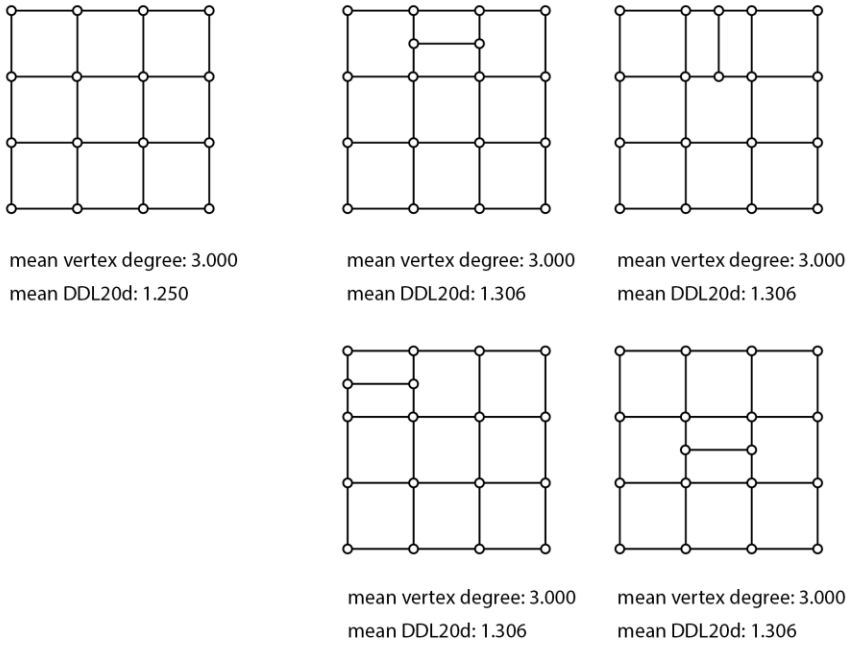

Figure 21. Non-equivalent street graphs yielded by applying the operation of linking edges exactly once on a 4 × 4 grid. Here, the final street graphs are the same as the intermediate ones.

2.3.9 Summary of effects

Based on the limited set of examples shown in the above section, the effect of each operation on the average vertex degree and the mean DDL20d of the street graph is summarized in *Table 3*.

As shown in Table 3, the effect of each operation is consistent when applied on a simple 4 × 4 square grid just once, as measured by the average vertex degree and the mean DDL20d. We have also proved that the operations of splitting vertex and shifting vertex have consistent impacts on the average vertex degree regardless of the

configuration of the original street graph, as long as there is at least one cell other than the wrapper cell in the original street graph.

Table 3. Changes in the mean vertex degree and the mean DDL20d after the first application of each operation on a 4 × 4 square grid

Name	Change in $\overline{deg}$		Change in mean *DDL20d*	
	Int.[a]	Final	Int.	Final
Shift vertex	No change	—	Increased	—
Contract edge	Increased	—	Increased	—
Cross-concatenate vertices	Decreased	—	Increased	Increased
Disjoin vertices	Decreased	Decreased	Increased	Increased
Split vertex	Decreased	—[b]	Increased	—
Link vertex to vertex	Increased	—	Increased	—
Link vertex to edge	Increased	—	Increased	—
Link edge to edge	No change	—	Increased	—

Note.

[a] Int. stands for Intermediate.

[b] The long dash indicates that the final street graphs are the same as the intermediate ones.

However, here we have only studied the effect caused by the first application of each operation. Will the effect be the same as we apply each operation more than once? How global structure of differentiation may arise from the incremental local variations? These questions will be studied in the following chapters.

CHAPTER 3
INTRODUCTION OF DATA STRUCTURE AND ANALYTICAL MEASURES

In Chapters 4–9, we will introduce different algorithms developed to generate superblock designs based on the syntactic operators introduced in the previous chapter. In this chapter, we first briefly introduce the data structure used in our generative algorithms to represent the street graphs. Then, we also explain some of the measures used to analyze the designs in the following chapters.

3.1 Data Structure Used for Implementing the Generative Algorithms

We used a *doubly-connected edge list* (DCEL) to store the interrelationships between vertices, edges, and cells. The DCEL is a well-known data structure used in many algorithms of computational geometry to represent planar subdivisions induced by planar embeddings of graphs (de Berg, Cheong, van Kreveld, & Overmars, 2008). The DCEL we implemented is essentially the same as explained in de Berg et al. (2008), with only minor variations. Below we use an example to briefly introduce the DCEL we used to represent the street graphs.

Figure 22 shows a street graph representing a 4 × 4 square grid. Each edge (represented by black lines) is associated with two *half-edges* (represented by red lines with arrows) lying on its two sides and pointing at opposite directions. The two half-edges associated with the same edge are called *twins*. Within each cell, the half-edges are oriented counterclockwise. The DCEL we used consists of four collections of records: one for the vertices, one for the half-edges, one for the edges, and one for the cells.

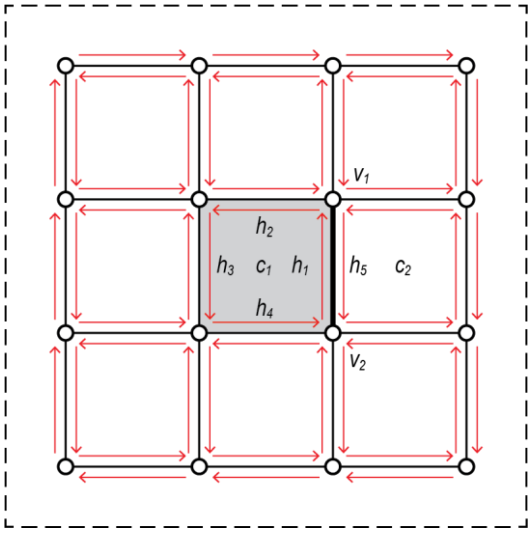

Figure 22. Representation of a street graph using DCEL.

For each vertex *v*, the following information is stored:

- *v.uid*: a unique ID assigned to *v*;
- *v.h*: a pointer to a half-edge arbitrarily selected from all the half-edges that originate from *v*; and
- *v.oref*: a pointer to the object that contains the geometric information about *v* such as the coordinates of *v*.

For example, assume that the vertex v_1 in Figure 22 is associated with the half-edge h_5, then h_5 can also be retrieved via calling $v_1.h$.

For each half-edge *h*, the following information is stored:

- *h.uid*: a unique ID assigned to *h*;
- *h.prev_h*: a pointer to the half-edge that leads to *h*;
- *h.next_h*: a pointer to the half-edge that *h* leads to;

- *h.twin_h*: a pointer to the twin of h;
- *h.e*: a pointer to the edge that *h* is associated with;
- *h.v*: a pointer to the vertex lying at the endpoint of *h.e* that *h* points to; and
- *h.c*: a pointer to the cell that is incident with and lies to the left of *h*.

We call the vertex that *h* points to, or *h.v*, the *target vertex* of *h*. We call the vertex which *h* originates from, or *h.twin_h.v*, the *origin vertex* of *h*. The cell that lies to the left of *h*, or *h.c*, is the cell which *h* belongs to.

For example, in Figure 22, the vertex v_1 is the target vertex of h_1 and can be retrieved by calling $h_1.v$. The vertex v_2 is the origin vertex of h_1 and can be retrieved by calling $h_1.twin\_h.v$, which essentially amounts to calling $h_5.v$ since h_5 is the twin of h_1 and can be retrieved by calling $h_1.twin\_h$. The half-edge h_2 can be retrieved by calling $h_1.next\_h$, and h_4 retrieved by calling $h_1.prev\_h$. The cell c_1 (shaded in Figure 22) is incident with h_1 and lies to its left, thus c_1 can be retrieved by calling $h_1.c$. The edge $\{v_1, v_2\}$ can be retrieved by calling $h_1.e$.

For each edge *e*, the following information is stored:

- *e.uid*: a unique ID assigned to *e*
- e.*h*: a pointer to any one of the two half-edges associated with *e*
- e.*guid*: a pointer to the object which contains the geometric information about *e*

For each cell *c*, the following information is stored:

- *c.uid*: a unique ID assigned to *c*
- *c.h*: a pointer to an arbitrary half-edge that belongs to *c*

- *c.is_wrapper*: a Boolean indicating whether the cell is the imaginary cell wrapping around the street graph. This is useful for checking whether an edge or a vertex lies on the boundary of a superblock design.

Note that in the above we have used the *dot* (.) as a shorthand for a function call to access the relevant information. The expressions involving the dot (.) should be evaluated from left to right. For example, as shown in Figure 22, the origin vertex of h_1 can be retrieved by calling h_1*.twin_h.v*, which in plain English can be interpreted as "get the target vertex of the twin of h_1." Similarly, by calling h_1*.next_h.next_h*, we first check the half-edge that h_1 leads toward, which is h_2, and then get the half-edge that h_2 leads to, which is h_3, as shown in Figure 22.

3.2 Introduction of Analytical Measures

In the following chapters, the designs generated with our algorithms are further analyzed from the following aspects: (a) graph property, (b) density, (c) directional reach and directional distance, (d) regularity, and (e) diversity. Specific measures are computed to analyze each of those aspects.

A particularly interesting and common dimension for the spacing of the arterial streets is 800 meters, or ½ mile, as can be seen in the U.S., China, and many other different urban contexts. Therefore, we will focus on the differentiated grids that share a consistent boundary condition—that is, a ½ mile by ½ mile square. Our generative algorithms always start with an initial street graph which resembles a 9 × 9 square grid with a side length of 800 m. Below we introduce those measures by computing those measures for the initial square-grid design.

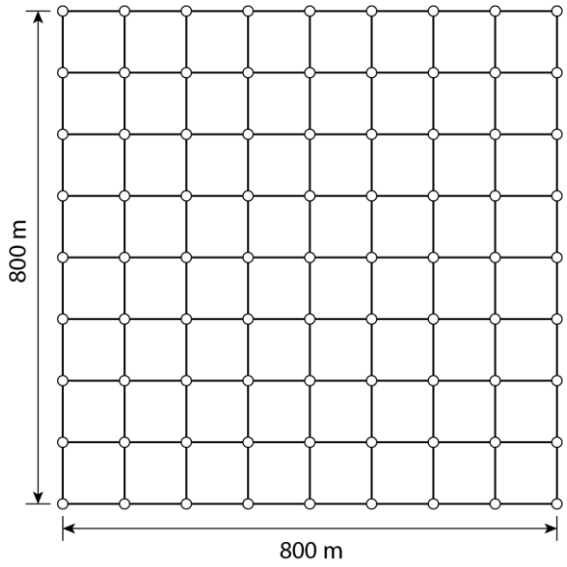

Figure 23. A 9 × 9 square-grid design.

3.2.1 Elementary graph properties

Four measures are included: (a) the total number of vertices per design, (b) the total number of edges per design, (c) the total number of cells per design, and (d) the mean vertex degree per design.

The initial 9 × 9 square-grid design has 81 vertices, 144 edges, and 64 cells. The 49 vertices inside the boundary of the superblock have degree 4, the 28 vertices located along the boundary but not the corners of the superblock have degree 3, and the four vertices sitting at the corners of the superblock have degree 2. Therefore, the sum of all vertex degrees for the initial design is $4 \times 49 + 3 \times 28 + 2 \times 4 = 288$, and the mean vertex degree is $288 / 81 \approx 3.556$.

In analyzing the designs, the measures related to the graph property are always evaluated based on the “unpolished” street graphs, while all the other measures are computed based on the “polished” street graphs.

3.2.2 Density of streets, blocks, intersections, and connectivity

Four measures are included: (a) the total length of streets per design, (b) the total number of blocks per design, (c) the total number of intersections per design, and (d) the mean distance between intersections per design.

There are in total 9 horizontal traversing streets and 9 vertical traversing streets in the initial design, each of which is 800 m long. Therefore, the total length of streets in the initial design is 800 × 14400 m. Since we do not count the imaginary wrapper cell in computing the total number of cells in a street graph, the total number of blocks is always equal to the total number of cells. Thus, there are 64 blocks in the initial design. The total number of intersections is equal to the total number of vertices with degree greater than 2. In the initial design, all vertices have degree greater than 2 except for the four vertices at the corners of the superblock. Therefore, the total number of intersections in the initial design is simply 81 – 4 = 77. The distance between two intersections is simply the length of the segment or the chain of segments (sometimes called a *road segment*) between the two intersections. In the initial design, the distance between intersections is uniformly 100 m except for the distance between the T-intersections near the corners of the superblock which is 200 m. Therefore, the mean distance between intersections for the initial design is (100 × 136 + 200 × 4) / 140 ≈ 102.86 m.

3.2.3 Directional reach and directional distance

Three measures are included: (a) the length-weighted mean directional distance per length, (b) the length-weighted mean of the linear reach (i.e., the zero-direction-change reach), and (c) the length-weight mean of the two-direction-change reach.

In calculating the directional distance, a threshold angle is set for determining whether a deviation in the direction of movement is large enough to be counted as a direction change. In the following analysis, unless otherwise specified, the threshold angle is always set to 20°. Moreover, in calculating the means for the directional distance and the directional reaches, we take into account the segment length so that longer segments contribute more "weight" than others in the calculation.

In the initial design, for any segment, the linear reach is 800 m. Since all the segments are of equal length, the length-weighted mean linear reach for the initial design is 800 m. The two-direction-change reach for any segment in the initial design is 14400 m (i.e., the total length of entire network) because it never takes more than two direction changes from any segment to any other segment in the initial design. Thus, the mean two-direction-change reach for the initial design is 14400 m. It is clear that wherever we are in the initial design, we can reach an entire horizontal or vertical traversing street without changing any directions. If we are allowed to walk everywhere without taking more than one turn from the original place, we can reach another nine traversing streets. If we are allowed to make two turns, then we can reach yet another eight traversing streets in addition to what we have already been able to reach. Therefore, the mean directional

distance per length for the initial design is (800 × 0 + 800 × 9 × 1 + 800 × 8 × 2) / 14400 ≈ 1.389.

The following abbreviations are used for the above measures: (a) *DDL20d*, which is short for the directional distance per length with the threshold angle set to 20°; (b) *dr0dc20d*, which is short for the linear reach with the threshold angle set to 20°; (c) *dr2dc20d*, which is short for the 2-direction-change reach with the threshold angle set to 20°.

3.2.4 Geometric regularity

Before introducing the measures used to calibrate the regularity of a design, we first introduce the notion of *continuity lines*. In the field of space syntax, the term "continuity lines" was introduced by Figueiredo and Amorim (2005) to indicate the quasi-linear lines resulting from the aggregation of axial lines. Our definition of the continuity lines is inspired by their work yet slightly different. The obvious difference is that we generate the continuity lines in different contexts. In our case, the continuity lines are created based on street graphs instead of axial maps. But a more important difference is that in generating the continuity lines, we not only check the geometric relationship but also the syntactic relationship between the segments in the street graph. Therefore, the continuity lines can only be generated after the directional distance analysis has been carried out. We illustrate this point by showing the algorithm of how we generate a set of continuity lines from a street graph. Suppose that as input to the algorithm we have a priority dictionary named *pqdict* which behaves much like a priority queue but with optional keys so that we can map the IDs of the edges of a street graph to their assigned

DDL values. The priority dictionary supports a procedure *pop*() which always gives us the ID of the edge which has the lowest DDL value in the dictionary *pqdict*. Initially, the dictionary *pqdict* contains key-values pairs which include all the edges and their DDL values of a street graph. We also use α to denote the threshold angle which is used to determine whether a deviation along a path is to be counted as a direction change or not.

Figure 24 shows a few continuity lines that are selected in order based on the above algorithm. As shown in Figure 24a, since the segments *AB* and *BC* have the least DDL value among all segments in the design, they are the first ones to be popped out from the priority dictionary to build continuity lines that go through them. Consequently, the first continuity line constructed by our algorithm is the line *A-B-C-D-E*. Since the next lowest DDL value in the design is assigned to the segment *HC*, the construction of the second continuity line starts from the segment *HC*. As shown in Figure 24b, at the point *C*, there are two segments whose deviation from the segment *HC* is smaller than the threshold angle ($\alpha = 20°$). According to the construction process of continuity lines described by Figueiredo and Amorim (2005), the segment *CK* will be selected to form the continuity line with *HC* because it involves a smaller angle of deviation when transitioning from *HC* to *CK*. Based on our algorithm, however, the segment *CJ* would be selected instead, because it has a lower DDL value than *CK* or, in standard space syntax terminology, more "integrated" than *CK*. Therefore, in addition to geometric alignment between segments, the continuity lines generated by our algorithm also reflect the syntactic relationship between segments.

ALGORITHM 3.1: Generate Continuity Lines

```
procedure generate_continuity_lines (pqdict, α)
L_c ≔ an empty list
while pqdict is not empty
        e ≔ pqdict.pop()
        C ≔ {e}
        /* search from one endpoint of e */
        current_h ≔ e.h
        H ≔ the set of half-edges that originate from current_h.v and the angle of deviation is
        smaller than or equal to α
        if H ≠ ∅ then
                next_h ≔ the half-edge in H with the least DDL
        else
                next_h ≔ NULL
        while next_h is not NULL and next_h.e ∉ C
                current_h ≔ next_h
                C ≔ C ∪ {current_h.e}
                if current_h.e is in pqdict then
                        pqdict.pop(current_h.e)
                H ≔ the set of half-edges originate from current_h.v and the angle of
                deviation is smaller than or equal to α
                if H ≠ ∅ then
                        next_h ≔ the half-edge in H with the least DDL
                else
                        next_h ≔ NULL
        /* search from the other endpoint of e */
        current_h ≔ e.h.twin_h
        H ≔ the set of half-edges that originate from current_h.v and the angle of deviation is
        smaller than or equal to α
        if H ≠ ∅ then
                next_h ≔ the half-edge in H with the least DDL
        else
                next_h ≔ NULL
        while next_h is not NULL and next_h.e ∉ C
                current_h ≔ next_h
                C ≔ C ∪ {current_h.e}
                if current_h.e is in pqdict then
                        pqdict.pop(current_h.e)
                H ≔ the set of half-edges originate from current_h.v and the angle of
                deviation is smaller than or equal to α
                if H ≠ ∅ then
                        next_h ≔ the half-edge in H with the least DDL
                else
                        next_h ≔ NULL
        append C to Lc
return Lc
```

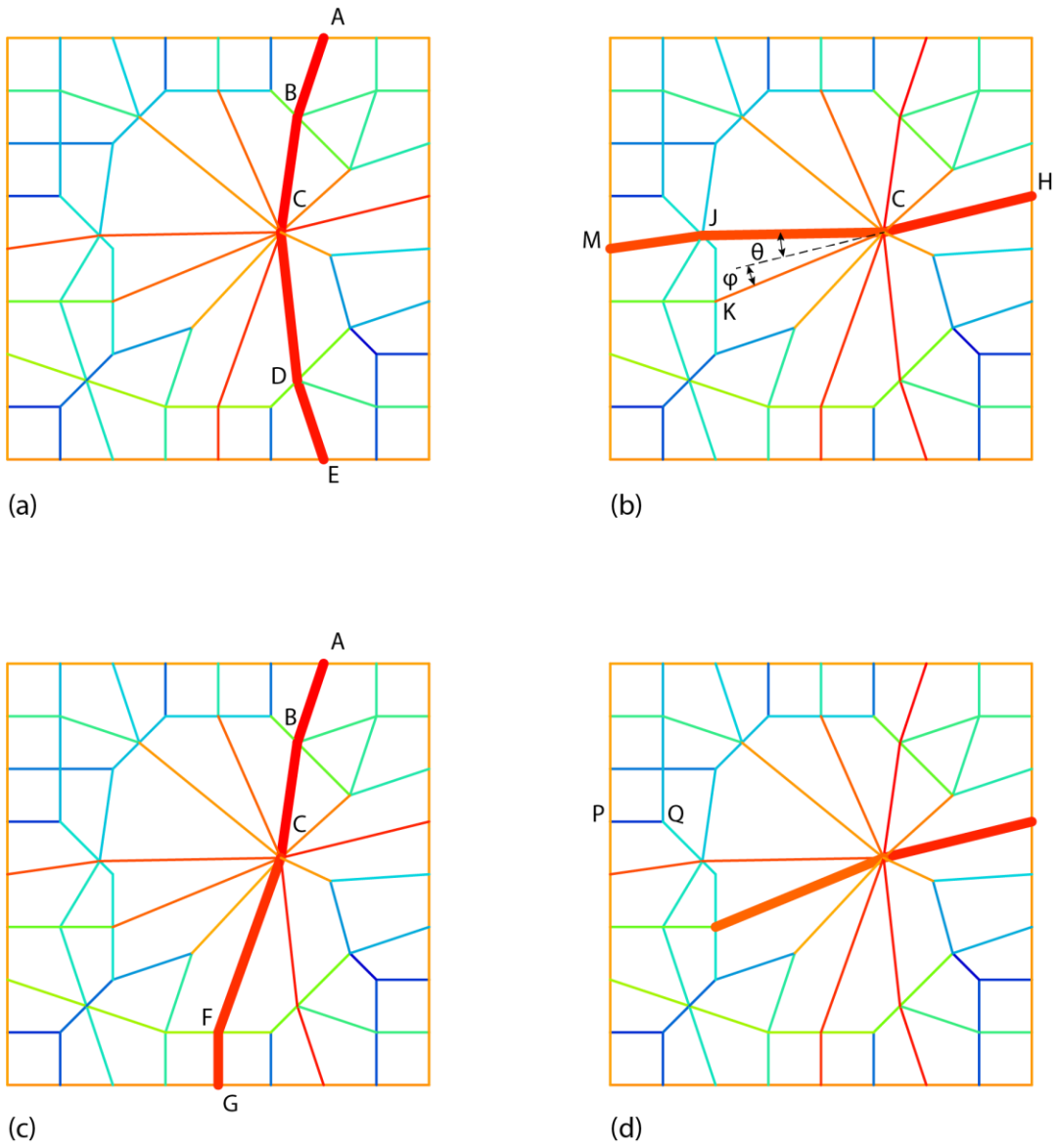

Figure 24. Constructing continuity lines for a hypothetical superblock design. The colors are assigned to the segments based on their assumed DDL values. The spectrum from red to blue corresponds to the range from low to high DDL values.

A continuity line may be made of a single segment—in other words, a segment may generate a continuity line that is only made of itself. It happens when moving from a segment to any segment immediately connected to it must involve an angle of deviation that is greater than allowed. For example, the segment *PQ* in Figure 24d would likely

make up a continuity line that is made only of itself. On the other hand, a segment can be reused to generate multiple continuity lines. As shown in Figure 24c, the segments *CF* and *FG* are selected to build the continuity line and the continuity line thus formed would include the segments *AB* and *BC* which have already been used to construct the first continuity line as shown in Figure 24a. For this reason, the sum of the length of all continuity lines generated by our algorithm could be greater than the total length of all segments in the design. The construction process continues until all the segments in the design have been used at least once. The DDL of a continuity line is defined as the length-weighted mean of the DDL values of the segments it comprises.

Four measures are included to calibrate the regularity or irregularity of a design: (a) fragmentality per design, which is defined as the ratio between the total number of continuity lines generated and the total number of segments in the design[15]; (b) the standard deviation of the block area per design; (c) the standard deviation of the block perimeter per design; (d) the mean standardized block area-perimeter ratio per design.

There are 18 continuity lines and 144 segments (i.e., edges) in the initial 9 × 9 square-grid design, thus the fragmentality for the initial design is 18 / 144 = 0.125. (A high degree of fragmentality suggests either many sharp turns along otherwise continuous streets or predominance of a dendritic street structure, thus suggesting an overall more

[15] If there are in total *n* segments in a design and they together form a single continuity line, then the fragmentality of the design is 1/*n*; if, however, each continuity line is made of only one segment, then the fragmentality of the design is *n*/*n* = 1. In other words, the fragmentality of a design composed of *n* segments varies between 1/*n* and 1.

deformed street design.) In the initial design, every block has the same area (10000 m^2) and perimeter (400 m). Therefore, the standard deviations of the block area and the block perimeter for the initial design both equal zero.

We interpret the perimeter of a cell or a block as the length of the footpath left by us when we walk along the boundary of the block and take a full circle back to where we were when we first started. Interpreted this way, we can simply measure the perimeter of a block by finding all the half-edges that belong to the corresponding cell and then summing the lengths of all the edges that are associated with those half-edges. For example, the street graph shown in Figure 25 contains two blocks, with one nested in the other. When computing the perimeter for the outer block, we sum the lengths of $h_5.e$, $h_6.e$, $h_7.e$, $h_8.e$, $h_9.e$, $h_{10}.e$, $h_{11}.e$, $h_{12}.e$, $h_{13}.e$, $h_{14}.e$, and $h_{15}.e$. Since $h_8.e$ and $h_{13}.e$ refer to the same edge (highlighted by a thicker line in the street graph on the left), the length of that edge is counted twice. Thus, the two blocks in that street graph are interpreted as the ones shown in Figure 25b. Likewise, the length of the cul-de-sacs—if there are any—in a block will be counted twice in calculating the perimeter of the block because a person has to retrace a cul-de-sac to get back to where he/she was before entering the cul-de-sac.

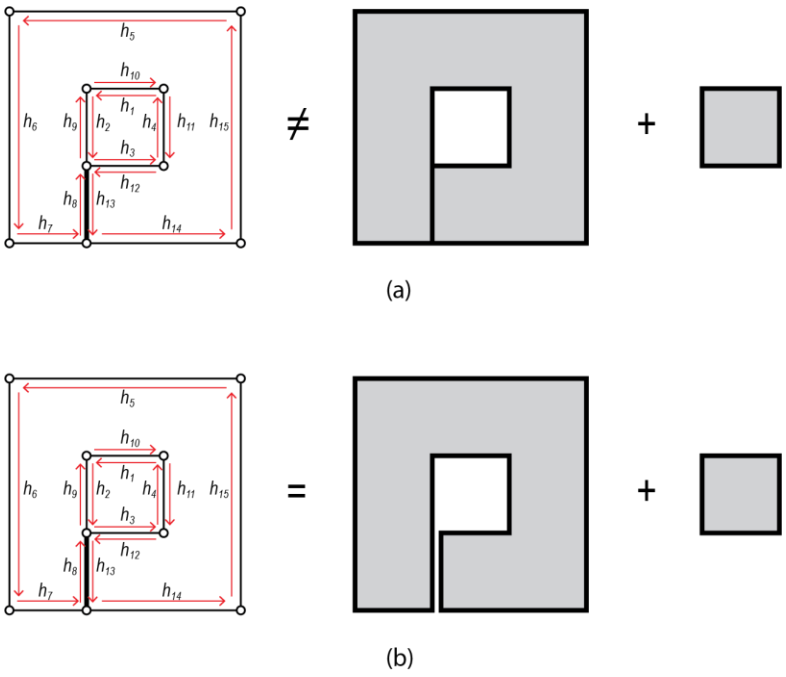

Figure 25. Interpretation of the perimeter of a block.

The area-perimeter ratio (APR) is often used to measure the "compactness" of a shape. Simply dividing the block area by its perimeter yields the APR for the block. However, it would be problematic to make direct comparisons based on APR, because the APR for a shape would change when the size of the shape changes even if the shape, regardless of the size, remains exactly the same. Figure 26 shows how the APR increases as a function of the side length of a square block: the APR is 25 for a square block whose side length equals 100 m, while the APR jumps to 100 for a square block whose side length equals 400 m. To address this issue, the APR is further standardized by comparing to the APR of a square block of the same size, and we call this measure standardized

block area-perimeter ratio (SAPR)[16]. Let a denote the area of a block and p denote the perimeter of the same block, then the SAPR of the block can be computed as $4\sqrt{a}/p$. In the initial design, all blocks are of the same shape and size, therefore have the same SAPR—that is, 1.00.

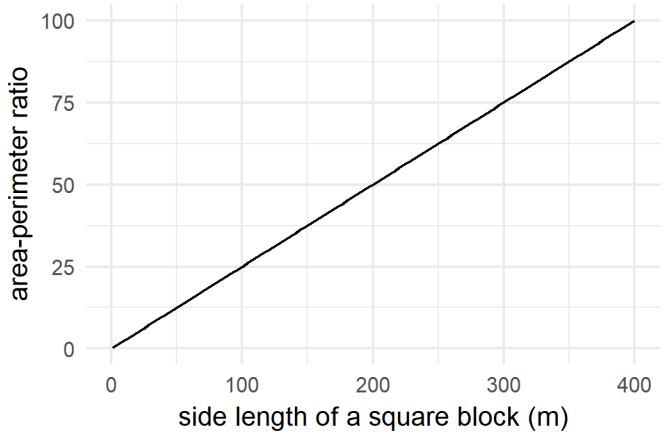

Figure 26. The area-perimeter ratio of a square block increases as a function of the side length of the square block.

3.2.5 Diversity in syntactic conditions

Three measures are included: (a) the total number of distinct DDL values observed in the design; (b) the proportion of distinct DDL values per design; and (c) the standard deviation (sd) of the DDL values per design.

[16] SAPR is defined similarly to one of the measures of compactness used by Batty (2001). However, the area-perimeter ratio here is standardized by comparing to the area-perimeter ratio of an idealized square of the same size instead of an idealized circle.

Since all the segments in the initial square-grid design take the same DDL value, the total number of distinct DDL values in the initial design is simply one. The proportion of distinct DDL values per design is computed by dividing the total number of distinct DDL values observed in the design by the total number of segments in the design. This measure may more truthfully reflect the diversity of syntactic conditions in a design because it takes the size of the design into account. For the initial design, the proportion of distinct DDL values is $1 / 144 \approx 0.0069$. Since all the segments in the initial design assume the same DDL value, the standard deviation equals zero—there is no diversity in syntactic conditions in the initial design.

CHAPTER 4
SHIFTING VERTEX: CREATING CURVILINEAR STREETS

4.1 Operation: Shift Vertex

4.1.1 Definition of operation

The operation of shifting vertex moves a vertex to a new position, and by doing so, it also moves the corresponding end of each incident edge to the new position (Figure 27).

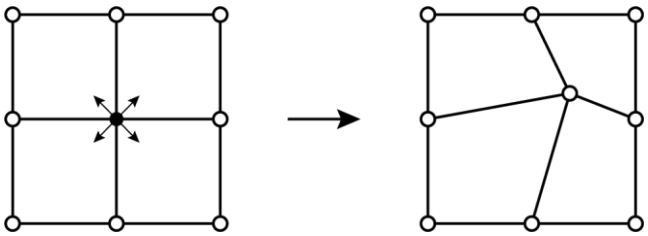

Figure 27. Shift a vertex.

4.1.2 Parameters

The operation of shifting vertex is parametrically controlled by the x- and y-coordinates of the displacement of the vertex (Figure 28).

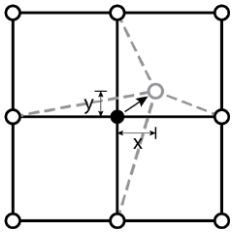

Figure 28. Parametric control of shifting vertex.

4.2 Generative Algorithm

4.2.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock designs, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by X;
- the number of streets running horizontally in the initial regular grid, denoted by Y;
- the vertex to be shifted, denoted by v;
- the maximum magnitude of the x- and y-axis components of the displacement of the vertex, denoted by d_{max};
- the minimum magnitude of the x- and y-axis components of the displacement of the vertex, denoted by d_{min};
- the minimum degree the vertex must have to be considered eligible for this operation, denoted by τ; and
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled via the following parameters: l_x, l_y, X, Y, v, d_{max}, d_{min}, τ, and T.

4.2.2 General description

We start from a street graph that represents a regular grid. Then, we randomly pick a vertex in the street graph and attempt the operation of shifting vertex. We continue to do so until the operation has been successfully performed a specified number of times. The attempted operation can only be successfully performed if *all* of the following conditions are met.

1. The randomly picked vertex, say v, has degree greater than or equal to the minimum degree, τ, required to perform the operation. Let $deg(v)$ denote the degree of v, this condition can be simply expressed as $deg(v) \geq \tau$.
2. The vertex v does not lie on the border of the street graph. One way to check this condition is to see if v is incident with the wrapper cell.
3. The intended position to be shifted to, indicated by a point p, lies completely within the cells that are incident with v.
4. After the intended shift, the order of the neighbors of v, sorted by their angle of rotation relative to the ray that starts out at p and passes through one of v's neighbors, say u, remains the same as before—when they were sorted by their angle of rotation relative to the ray that starts out at v and passes through u.
5. After the intended shift, for each vertex u_i that is adjacent to v, the order of the neighbors of u_i, sorted by their angle of rotation relative to the ray that starts out at u_i and passes through p, remains the same as before—when they were sorted by their angle of rotation relative to the ray that starts out at u_i and passes through v.
6. The intended shift does not lead to any unexpected intersections among edges.

While most of the above conditions are intended to control certain aspects of the design output, Conditions 4 and 5 are included only to simplify the task of maintaining the integrity of the underlying data structure. Below illustrate a few situations where some of the above conditions are not met (Figure 29). Note that the situations illustrated here are hypothetical and may be unlikely or impossible to occur by only applying the operation of shifting vertex on a regular grid. Nevertheless, we still check those conditions here so that the proposed algorithm can be used in more general situations.

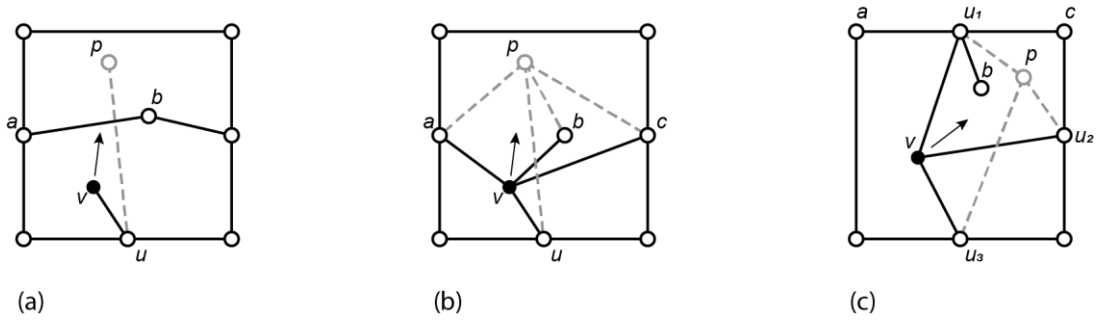

Figure 29. Examples of illegitimate moves (shift vertex).

In the example shown in Figure 29a, the intended shift of the vertex v is not allowed because the intended position to be shifted to, marked by the point p, does not lie in the cell incident with v—that is, a violation of Condition 3, and also because the edge $\{p, u\}$ would intersect the existing edge $\{a, b\}$—that is, a violation of Condition 6. Of course, if we decide that a vertex must have degree greater than 1 to be considered eligible for this operation, or set $\tau > 1$, then v should not be considered in the first place since $deg(v) < \tau$—that is, a violation of Condition 1.

In the example shown in Figure 29b, the order of the neighbors of *v* before the intended shift is *u*, *a*, *b*, *c*, sorted by their angle of rotation (counterclockwise) relative to the ray that starts out at *v* and passes through *u*. After the intended shift, the order of the neighbors of *v* becomes *u*, *a*, *c*, *b*, sorted by their angle of rotation (counterclockwise) relative to the ray that starts out at *p* and passes through *u*. Since there is a change in order, it violates Condition 4.

In the example shown in Figure 29c, although the order of the neighbors of *v* would not change as required by Condition 4, the order of the neighbors of u_1, one of *v*'s neighbors, would change after the intended shift. Before the intended shift, the order of the neighbors of u_1 is *v*, *b*, *c*, *a*, sorted by their angle or rotation (counterclockwise) relative to the ray that starts out at u_1 and passes through *v*. After the intended shift, however, the order of the neighbors of u_1 would change to *v* (*as marked by p*), *c*, *a*, *b*, sorted by their angle of rotation (counterclockwise) relative to the ray that starts out at u_1 and passes through *p*. Thus, it violates Condition 5.

4.2.3 Pseudocode

Suppose that we have already generated the initial street graph *G* which represents a regular grid that is l_x units long, l_y units wide, with *X* number of streets running vertically and *Y* number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of shifting vertex based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random* (*a*, *b*) refers to a procedure which returns a random floating point number *N* such that $a \leqslant N \leqslant b$ for $a \leqslant b$, and *random_choice* (*seq*) refers to a

procedure which returns a random element from the sequence *seq*. Condition 4 is simply stated as "the angular order of $N(v)$ remains the same," where $N(v)$ denotes the set of all neighbors of v, or the neighborhood of v; Condition 5 is simply stated as "for each u in $N(v)$, the angular order of $N(u)$ remains the same." Moreover, we define a point p by a position vector (x, y), and we use $v.x$ and $v.y$ to reference the x- and y-coordinates of the original position of the vertex v.

ALGORITHM 4.1: Shift Vertices

```
procedure shift_vertices (G, d_max, d_min, τ, T)
counter ≔ 0
while counter < T
        v ≔ a randomly picked vertex from the vertex set V of G
        if v does not lie on the border of G and deg(v) ≥ τ then
                x ≔ random (d_min, d_max) × random_choice ([1, -1])
                y ≔ random (d_min, d_max) × random_choice ([1, -1])
                p ≔ (v.x + x, v.y + y)
                if p lies completely within the incident cells of v and
                after v is shifted to the position of p,
                the angular order of N(v) remains the same
                and for each u in N(v), the angular order of N(u) remains the same
                and it does not lead to any unexpected intersections among edges then
                        shift v to the position of p
                        counter ≔ counter + 1
```

4.3 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the six groups of designs, we varied only one control parameter—namely T, the total number of operations to be performed—but kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9, d_{min} = 9 m, d_{max} = 26 m, τ = 2.

4.3.1 A note on the minimum and maximum displacements allowed

Shifting a vertex on a regular grid would cause a deviation from a straight course because of the bent street. The maximum deviation occurs when the vertex is shifted to the greatest extent possible along both the x- and y-axes. Take for example the grid shown in Figure 30, the greatest deviation angle caused by the shifting of the vertex *v* from *B* to *F* is the angle shaped by the segments *DF* and *FA*, labeled by α. To study the relationship between the magnitude of the displacement along the x- and y-axes, *d*, and the deviation angle, α, we introduce the angle formed by the segments *BA* and *AF*, labeled as θ_1, and the angle formed by *AD* and *DF*, labeled as θ_2.

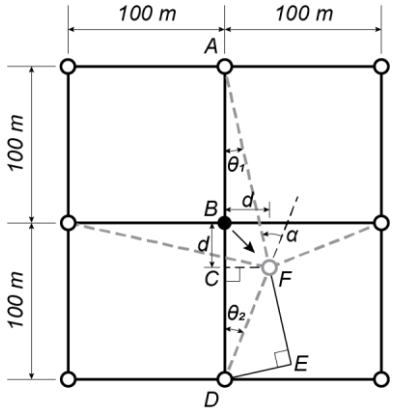

Figure 30. Deflection of streets caused by the operation of shifting vertex.

First, we observe that

$$\alpha = \theta_1 + \theta_2 \tag{9}$$

Second, since ΔAFC is a right triangle and the streets are placed at 100 m intervals, we know that

$$d = (100 + d)\tan\theta_1 \quad (10)$$

Finally, note that the segment *DF* is the common hypotenuse of two adjacent right triangles, namely Δ*DFC* and Δ*DFE*. Therefore,

$$(100 - d)\sec\theta_2 = 200\sin\theta_1 \csc\alpha \quad (11)$$

Based on the above equations, if α = 10°, then $d \approx 8.68$ m. Rounding it up to the nearest integer, we set d_{min} = 9 m, so that the angle of deviation along the street being bent on the first application of the operation of shifting vertex will certainly exceed 10°. If α = 30°, then $d \approx 25.20$ m. Rounding it up to the nearest integer, we set d_{max} = 26 m, so that the angle of deviation along the street being bent on the first application of the operation of shifting vertex could go up to 30°.

4.3.2 Data analysis

Six groups of designs were generated by applying the operation of shifting vertex 8, 16, 24, 32, 40, and 49 times, respectively. Each group consists of 100 designs. Theoretically, the operation of shifting vertex can be applied an arbitrary number of times on a regular grid based on the previously set parameters and the algorithm we developed. However, as we apply the operation more than one or two hundred times, the street graphs thus generated become very unrealistic and very hard to interpret in a meaningful way. Therefore, we tentatively set the upper limit for the number of times of application of this operation to be 49, which is the total number of vertices inside the boundary of the initial street graph (so that in the extremely unlikely case, each vertex in the initial street graph can be shifted once).

The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of the designs. Examples of designs from each group are shown in Figure 31. Note that the blocks (cells) consist of only quads.

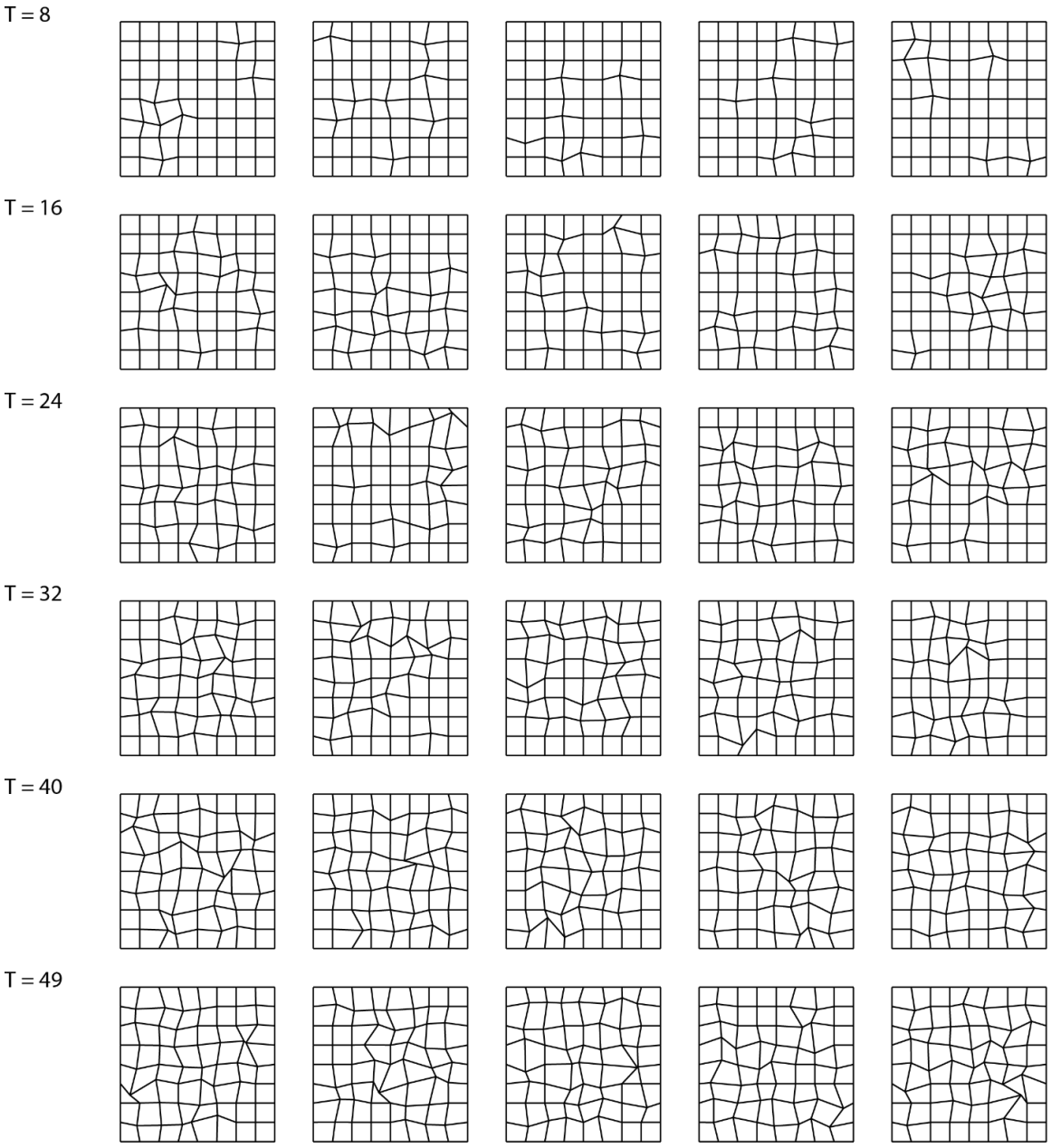

Figure 31. Examples of designs from each group (shift vertex).

Elementary graph properties

As algorithmically defined above, the operation of shifting vertex only varies the geometric properties of the street graph and does not add or remove any vertex, edge, or cell. Therefore, no matter how many times we apply this operation, the following are always true for each design: $|V| = 81$, $|E| = 144$, $|C| = 64$. The mean vertex degree per design is 3.556.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

As shown in Figure 32, the total street length for the superblock designs generated by shifting vertices alone is always greater than that of the initial 9 × 9 square-grid design, no matter how many times the operation has been applied in the generative process. Among all designs, the total street length per design could reach more than 14900 m—about a 3.5% increase in the total street length compared to the initial street graph.

If we plot the distribution of total street length per design for each group of designs, it becomes clear that the total street length for a superblock design tends to increase more as the operation is applied more frequently (Figure 32A). Moreover, the total street length per design tends to vary more and cover a wider range of values for the groups of designs generated by more frequent application of the operation of shifting vertex (Figure 32).

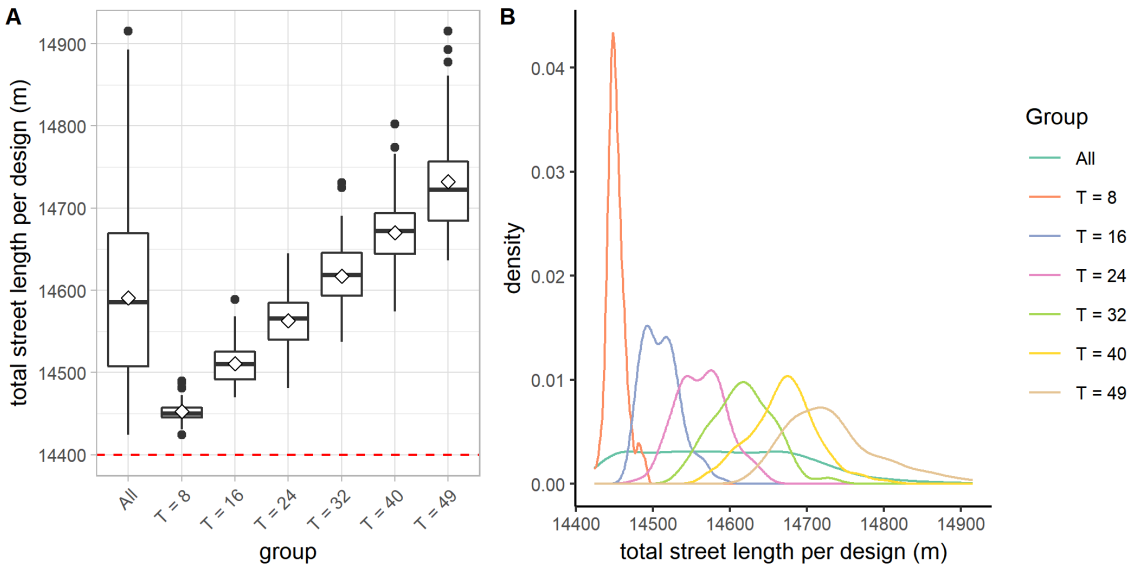

Figure 32. Distribution of the total street length per design (shift vertex). A: A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs. The red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m). B: Kernel density curves showing the distribution of the total street length per design for all designs and for each group of designs.

(2) Total number of blocks per design

Since shifting vertex alone does not lead to consolidation or subdivision of blocks, the total number of blocks in each generated superblock design is 64, the same as in the initial 9 × 9 square-grid design.

(3) Total number of intersections per design

Since shifting vertex alone does not add or remove any intersections, the total number of intersections in each generated superblock design is 77, the same as in the initial 9 × 9 square-grid design. Note that the total number of intersections is not equal to the total number of vertices in a street graph. Only vertices that have degree greater than

or equal to three are counted as intersections. Since the four vertices at the corners of the initial street graph have degree two, they are not counted as intersections in this analysis.

(4) Distance between intersections

As shown in Figure 33, while the distance between two intersections covers a wide range of values—from below 10 m up to more than 220 m, it is typically between 100 and 110 m. The mean distance between intersections per design, however, tends to increase as more vertices are shifted during the generative process. In fact, the distribution of the mean distance between intersections per design is identical to that of the total street length per design. This is hardly surprising because, for a superblock design without any cul-de-sacs, the mean distance between intersections is equal to the mean road segment length. Since the total number of road segments in a superblock design stays the same no matter how many times we apply the operation of shifting vertex, the average length of the road segments, or the average distance between intersections, is simply a function of the total street length in the design.

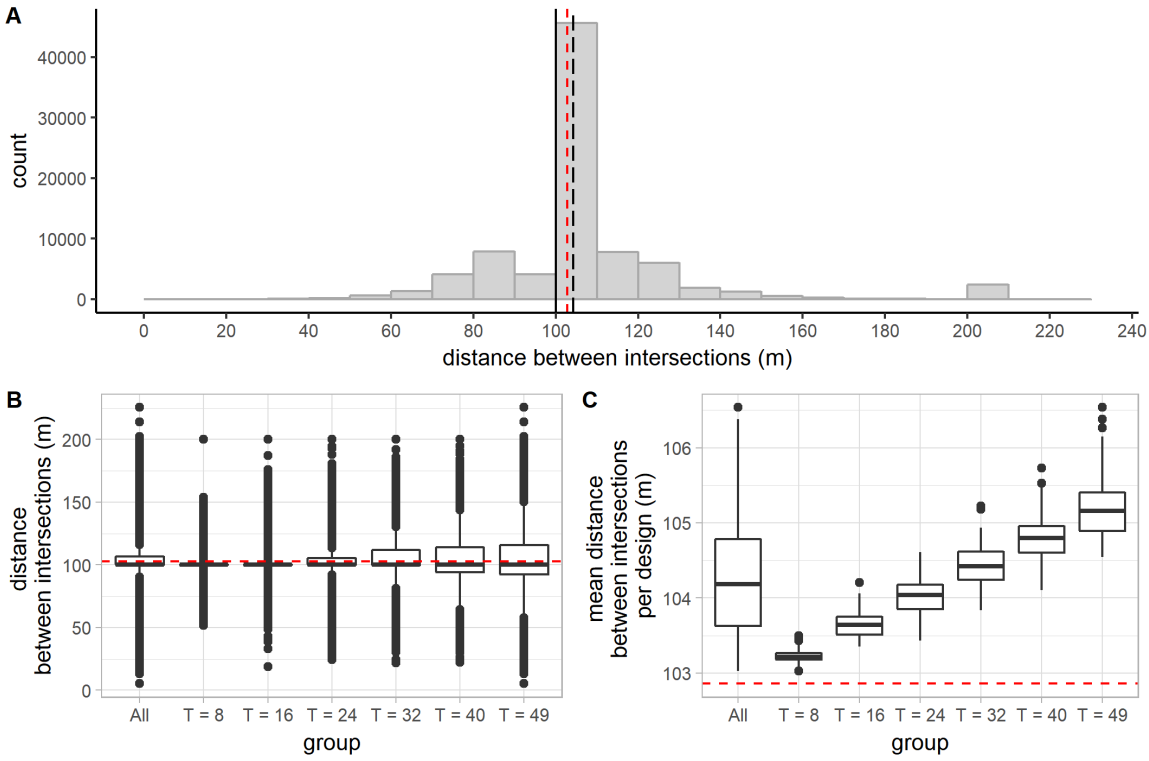

Figure 33. Distribution of the distance between intersections (shift vertex). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the distance between intersections observed in all designs and in each group of designs. C: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Directional reach and directional distance

(1) DDL (directional distance per length)

The values of DDL are dependent on how we count direction changes. As shown in Figure 34, changing the threshold angle (i.e., the minimum degree of deviation over which we count as a direction change) significantly changes the distribution of DDL values assumed by the segments in all designs. As we increase the threshold angle, the

DDL values in general become smaller. In fact, the mean DDL for all segments drops more than 30% as the threshold angle is increased from 10° to 30°. Moreover, the distribution of the DDL values is increasingly skewed to the right, with almost half of the segments assume a value lower than 1.5 as the threshold angle is brought up to 30° (Figure 34C). In the following analysis, unless otherwise specified, the directional distance is always evaluated by setting the threshold angle to 20°.

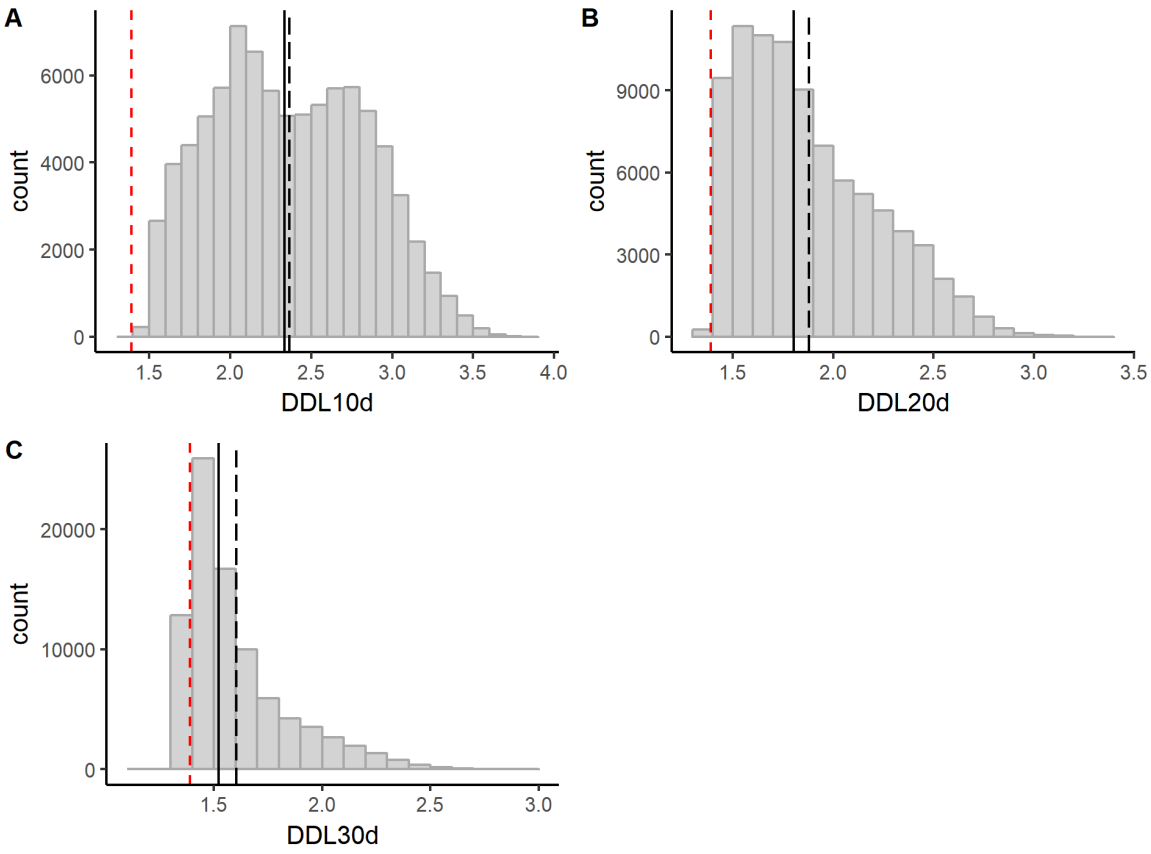

Figure 34. Distributions of DDL10d, DDL20d, and DDL30d (shift vertex). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

As shown in Figure 35A, the values of DDL20d for all segments range from slightly below 1.5 to nearly 3.5. As the data is heavily right-skewed, over eighty percent of all segments assume a DDL20d value lower than 2.2. In general, more frequent shifting of vertices tends to produce more extremely segregated segments. Similarly, as Figure 35B shows, the mean DDL20d per design tends to increase as the operation of shifting vertex is applied more frequently. Moreover, even the highest mean DDL20d in the group of designs with T = 8 is lower than the lowest DDL20d in the groups with T = 32, 40, and 49. In other words, the worst design from the point of view of mean DDL20d among the group of designs with T = 8 is still better than the best designs from the groups of designs generated by applying the operation of shifting vertex more than 30 times.

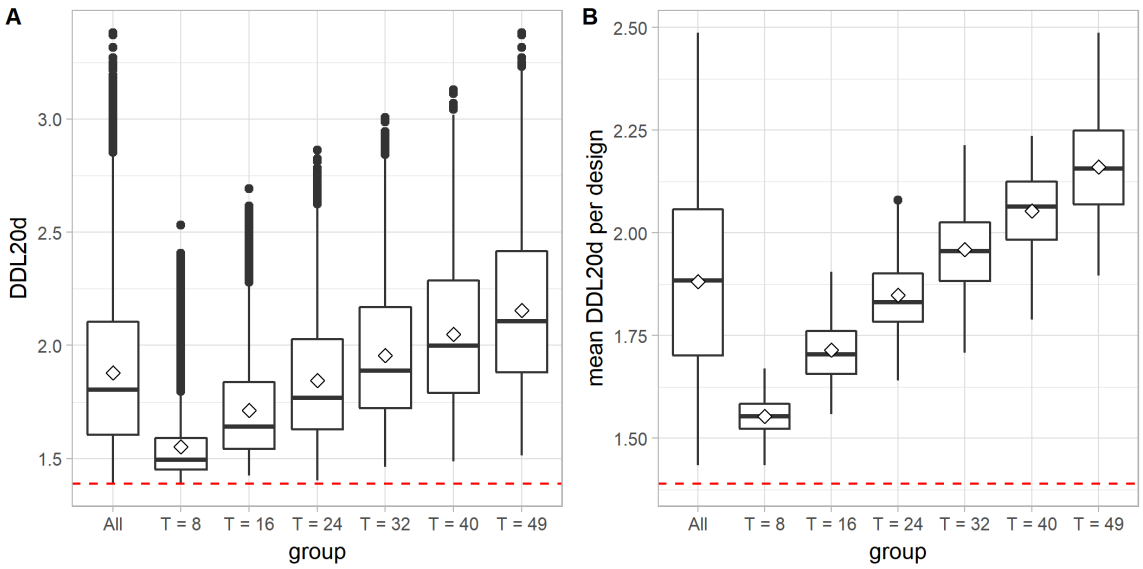

Figure 35. Distribution of DDL20d (shift vertex). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 36A, half the segments have a linear reach of more than 600 m, and forty percent of all the segments have a linear reach of between 800 and 850 m—that is, about the same as the width of the entire superblock. In other words, forty percent of all the segments lie on streets (either internal streets or bounding streets) that traverse the whole block. On the other hand, the linear reach values for the rest of the segments are spread almost evenly across the range between 0 and 800 m, with the least linear reach being as short as 14 m.

While the linear reach for the segments in each group of designs covers a similar range of values, there is a clear tendency for the mean linear reach to drop as the operation of shifting vertex is applied more frequently. The middle 50% of linear reach values are more dispersed—in a sense, more varied—as vertices are more frequently shifted, demonstrated by the increasingly widening inter-quartile ranges shown in Figure 36B. Similarly, the mean linear reach per design tends to drop as the operation of shifting vertex is applied more frequently. The mean of the mean linear reach per design for the group of designs with T = 49 is only 450 m, about a 35% drop compared to the mean for the group of design with T = 8, and an over 40% drop compared to the mean linear reach for the initial 9 × 9 square-grid design (which is 800 m).

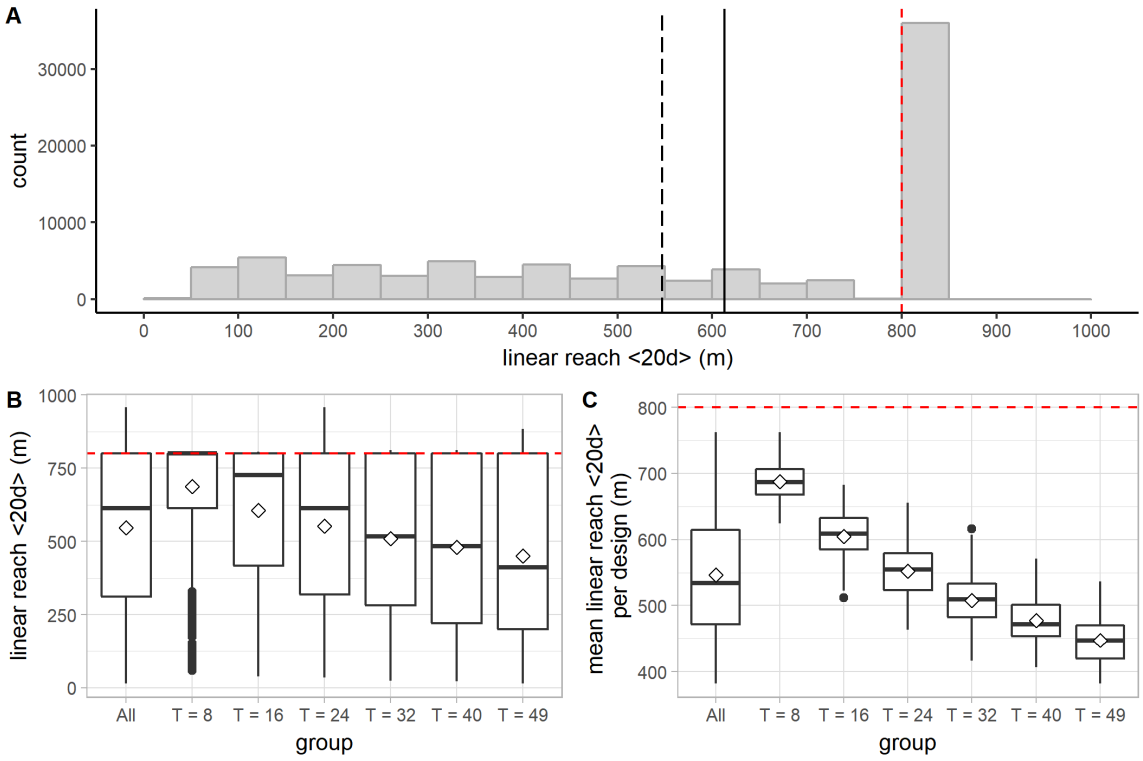

Figure 36. Distribution of the linear reach (shift vertex). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

(3) 2-dc reach (dr2dc20d)

As shown in Figure 37A, the distribution of 2-dc reach for all segments in all designs is heavily left-skewed. More than half the segments have a 2-dc reach greater than 12000 m, covering more than 80% of the entire street network in a superblock design. While most of the segments assume a good 2-dc reach, the least 2-dc reach for a segment could go below 3000 m.

When analyzed in groups, the 2-dc reach for the segments in the groups of designs generated by more frequent application of the operation of shifting vertex tends to cover a wider range of values, mostly pushing toward the lower end. For example, the lowest 2-dc reach in the group with T = 49 is only about half of the lowest 2-dc reach in the group with T = 8 (Figure 37B). Notice that the tendency in creating syntactically more segregated segments (places) is better captured in the distribution of DDL and 2-dc reach rather than that of the linear reach, mainly because the former two measures consider both direct and indirect relations, while the latter measure captures only immediate, local relations. Moreover, the middle 50% of 2-dc reach values are more dispersed (thus indicating greater variability) for the segments in the groups of designs generated by shifting vertices more frequently.

Similar to the mean linear reach per design, the mean 2-dc reach per design is likely to be lower for designs generated by more frequent shifting of vertices. It becomes clear by observing that even the worst design in terms of the mean 2-dc reach in the group of designs with T = 8 is still better than the best designs in the groups of designs with T = 32, 40, and 49 (Figure 37C). However, there tends to be greater variation in the mean 2-dc reach per design for the groups of design generated by more frequent application of the operation of shifting vertex.

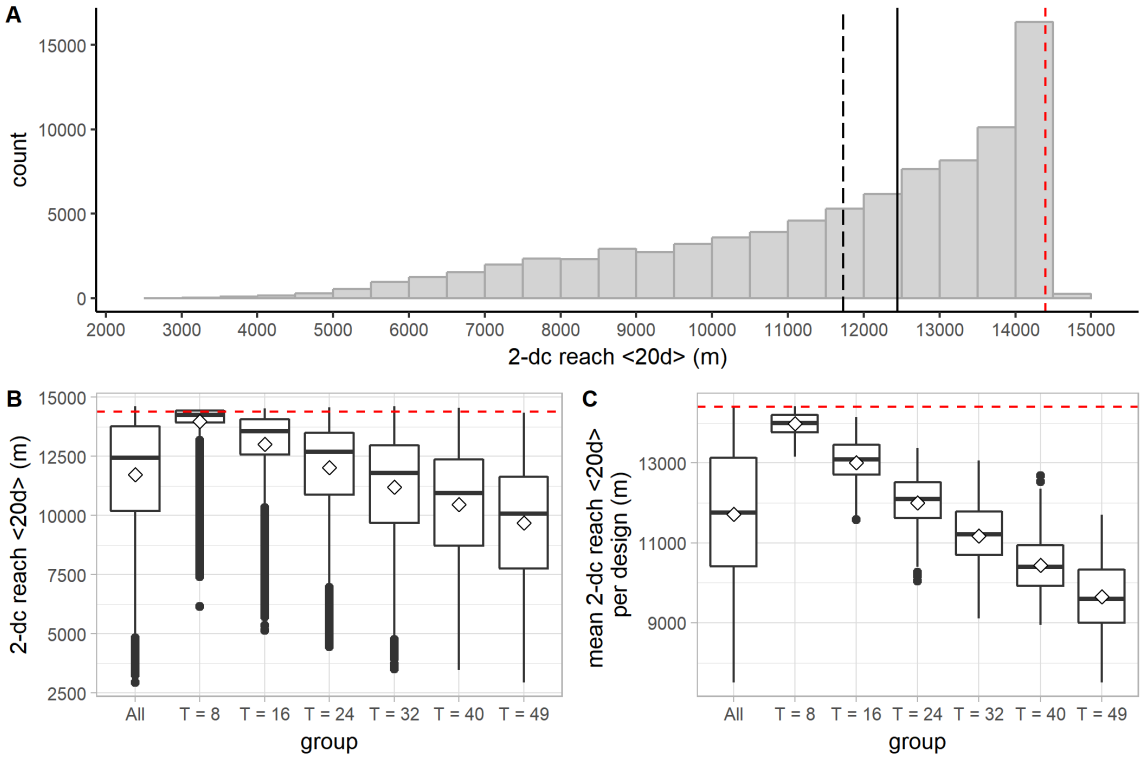

Figure 37. Distribution of the 2-dc reach (shift vertex). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

As shown in Figure 38, the fragmentality per design ranges from below 0.15 to nearly 0.5. The fragmentality per design tends to increase as the operation of shifting vertex is applied more frequently. Note that the most deformed design in the group with

T = 8 is still more regular than the most regular designs in the groups with T = 32, 40, and 49 (Figure 38B).

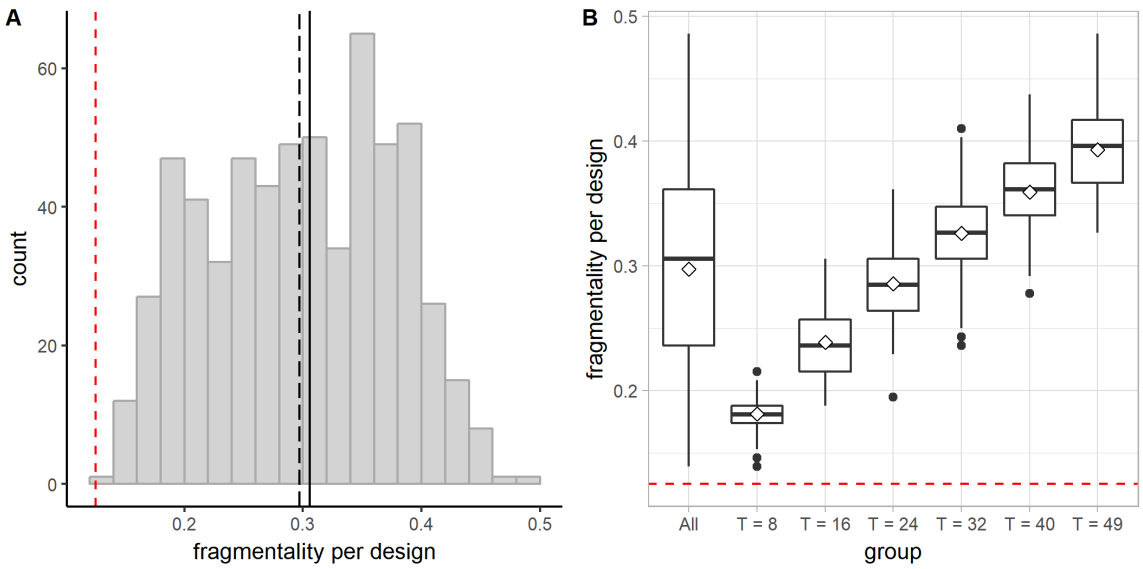

Figure 38. Distribution of the fragmentality per design (shift vertex). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

As shown in Figure 39, the area of a block ranges from below 2000 m² (the area of a fifth of a 100 × 100 m block) to slightly over 25000 m² (the area of two and a half 100 × 100 m blocks). When analyzed in groups, the block area tends to be more varied as vertices are shifted more frequently, as evidenced by the increasingly wider inter-quartile ranges (Figure 39B). It should be noted that although the area for individual blocks in a design can differ wildly, the mean block area per design, however, always stays the same

(10000 m²), because the total number of blocks stays the same under the operation of shifting vertex.

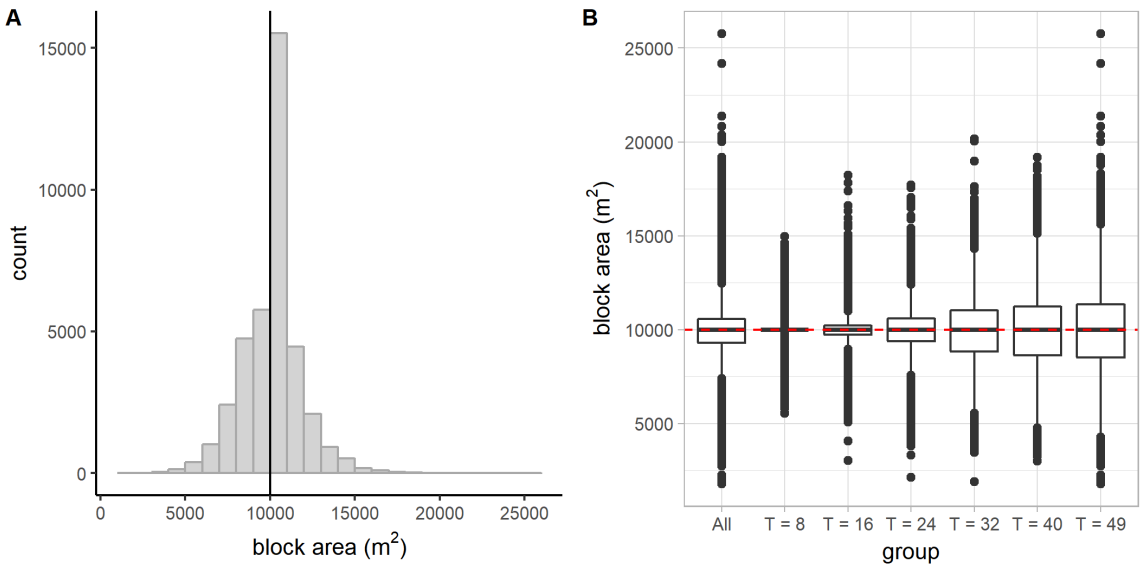

Figure 39. Distribution of the block area (shift vertex). A: A histogram showing the distribution of the area for all the blocks in all designs. The solid black line indicates the median. B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs. The red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²).

While the mean block area per design stays the same throughout the generative process, the areas of the blocks become more and more varied as the operation is applied more frequently, as evidenced by the increasing standard deviation of the block area per design (Figure 40).

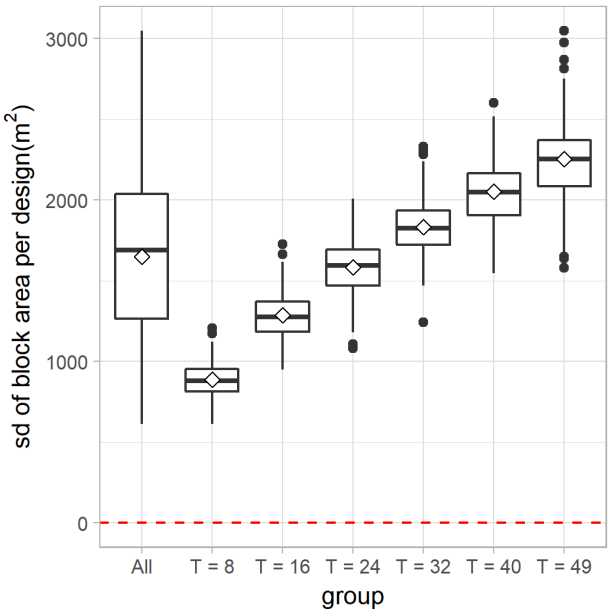

Figure 40. A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs (shift vertex). The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²).

(3) Block perimeter

As shown in Figure 41, the perimeter of a block ranges from 250 m to slightly over 650 m. The block perimeter is typically between 400 and 410 m (almost identical to the perimeter of a block in the initial 9 × 9 square-grid design) for all blocks in all designs. Similar to the block area, when analyzed in groups, the block perimeter tends to be more varied and their values wider apart as more vertices are shifted, as evidenced by the increasingly wider inter-quartile ranges (Figure 41B). Unlike the mean block area, however, the mean block perimeter per design does not stay the same for different designs. Instead, it tends to increase for the designs generated by more frequent shifting of vertices. In fact, the distribution of the mean block perimeter per design is identical to

that of the total street length per design. This is hardly surprising because the mean block perimeter for a design is a function of the total street length in the design—it can be roughly computed by dividing twice the total street length by the total number of blocks in the design.

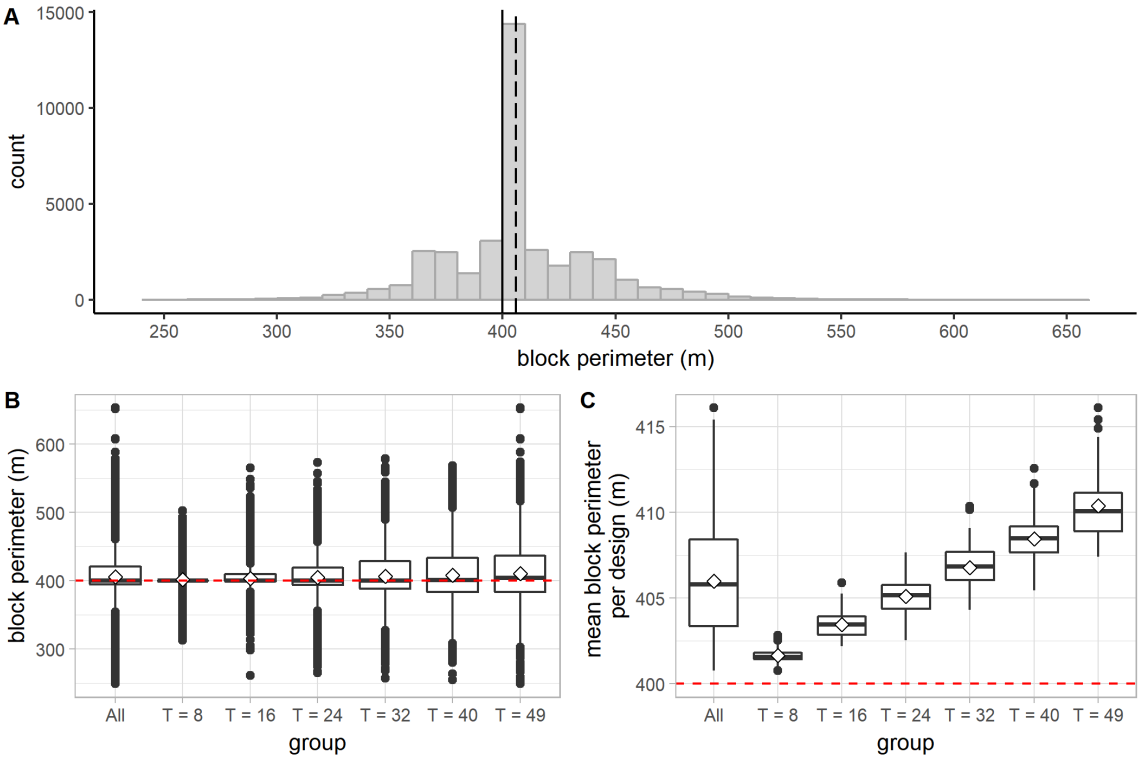

Figure 41. Distribution of the block perimeter (shift vertex). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. In the subfigure and the following one, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

The perimeters of blocks inside a design become more and more varied as the operation is applied more and more frequently, as evidenced by the increasing standard deviation of the block perimeter per design (Figure 42).

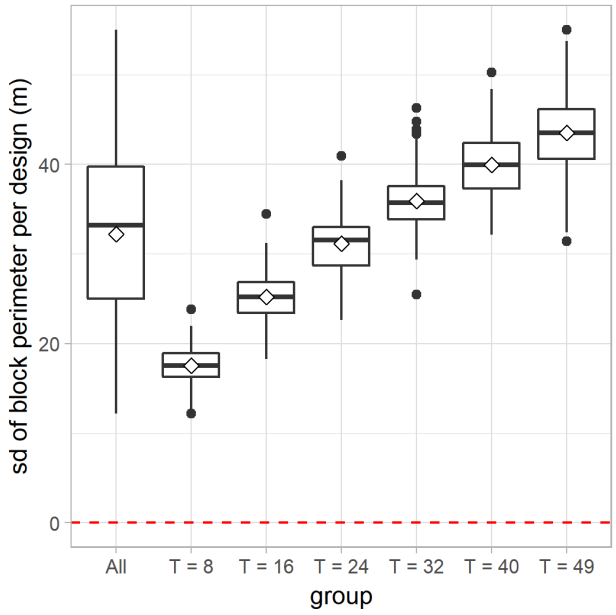

Figure 42. A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs (shift vertex). The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m).

(4) Standardized block area-perimeter ratio (SAPR)

As shown in Figure 43, about half of all the blocks generated have an SAPR greater than 0.99. In other words, half the blocks closely resemble a perfect square. When analyzed in groups, it is clear that as we apply the operation of shifting vertex more frequently, we are more likely to produce oddly shaped blocks—for instance, the lowest SAPR we get from the group with T = 49 is only slightly above 0.5, while the lowest SAPR we get from the group with T = 8 is still above 0.85 (Figure 43B). The mean SAPR per design tends to decrease for the groups of designs generated by more frequent shifting of vertices (Figure 43C).

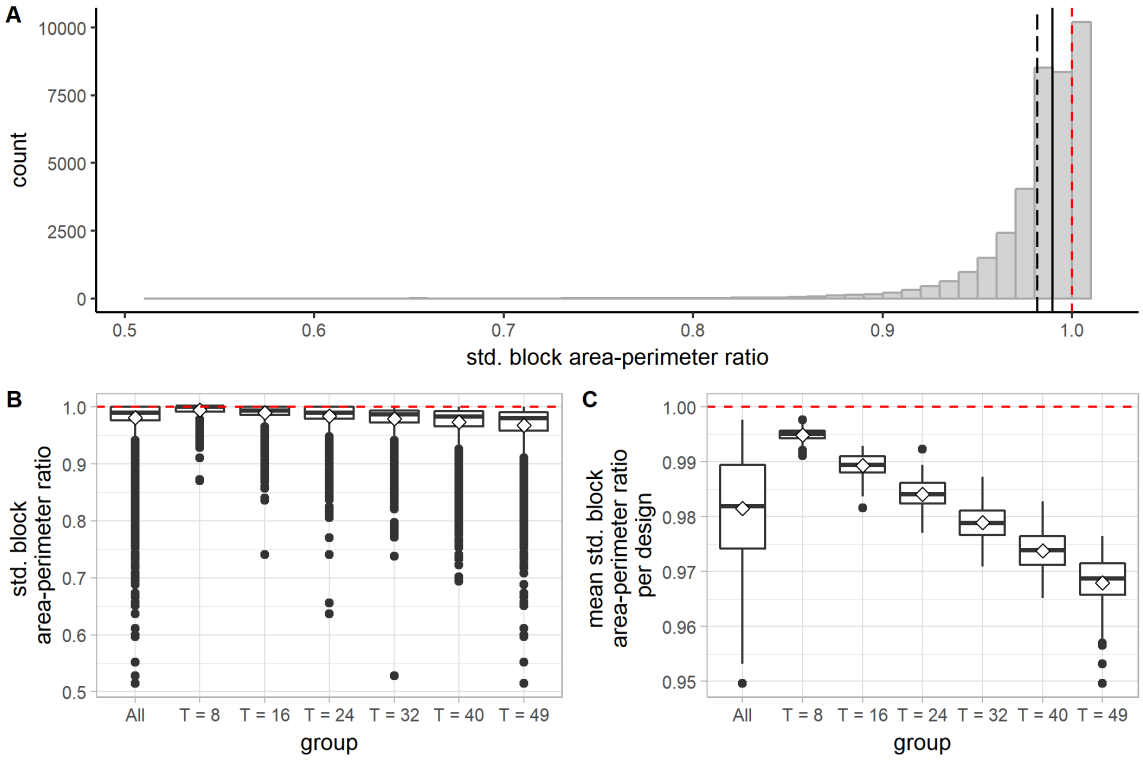

Figure 43. Distribution of the standardized block area-perimeter ratio (SAPR) (shift vertex). A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and proportion of distinct DDL20d values per design

One way to measure the diversity of syntactic conditions in a design is simply to count the total number of distinct DDL20d values assumed by the segments in the design. As shown in Figure 44A, the total number of distinct DDL20d values ranges from 15 to 70, a significant increase compared to the initial squared-grid design in which all

segments assume exactly the same DDL value. Half of the designs have more than 40 unique DDL20d values. The number of distinct DDL20d values per design tends to increase for the designs generated by more frequent application of the operation of shifting vertex (Figure 44B). Since shifting vertex does not change the total number of segments in a design, the proportion of distinct DDL20d values per design follows an identical trend (Figure 44C).

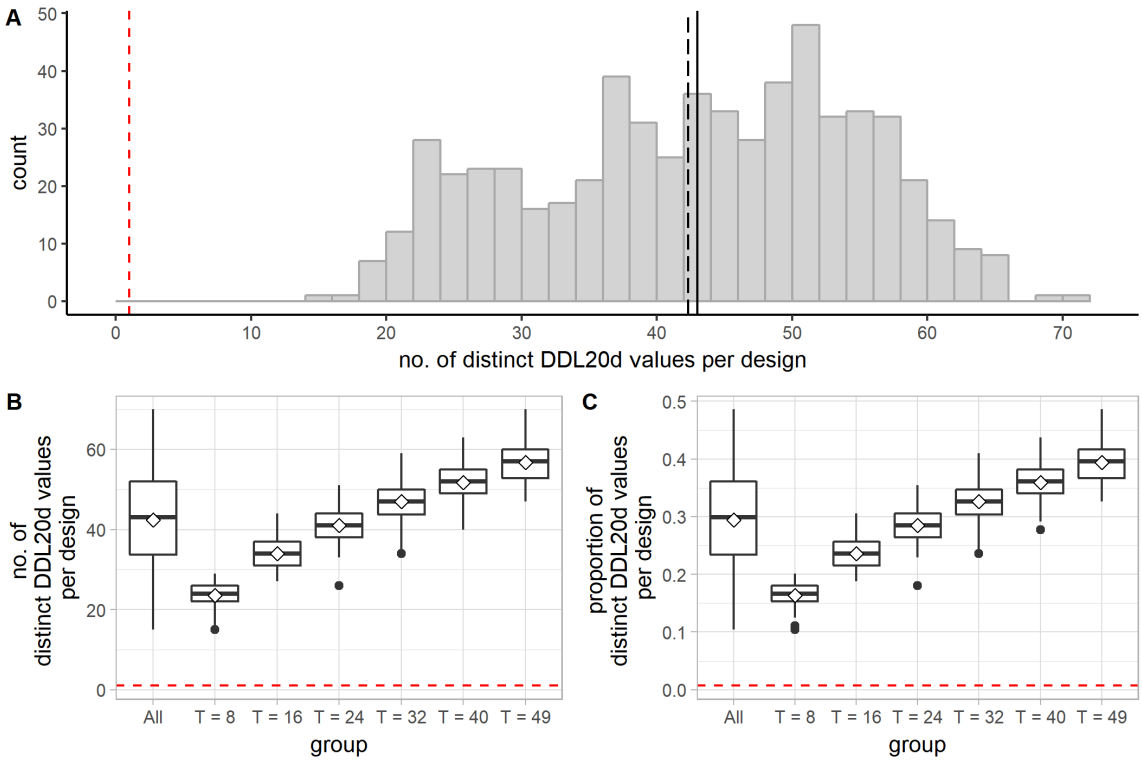

Figure 44. Distribution of the total number and the proportion of distinct DDL20d values per design (shift vertex). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d per design

As shown in Figure 45, the standard deviation (sd) per design ranges from below 0.1 to nearly 0.4. It tends to increase for the designs generated by more frequent application of the operation of shifting vertex (Figure 45B).

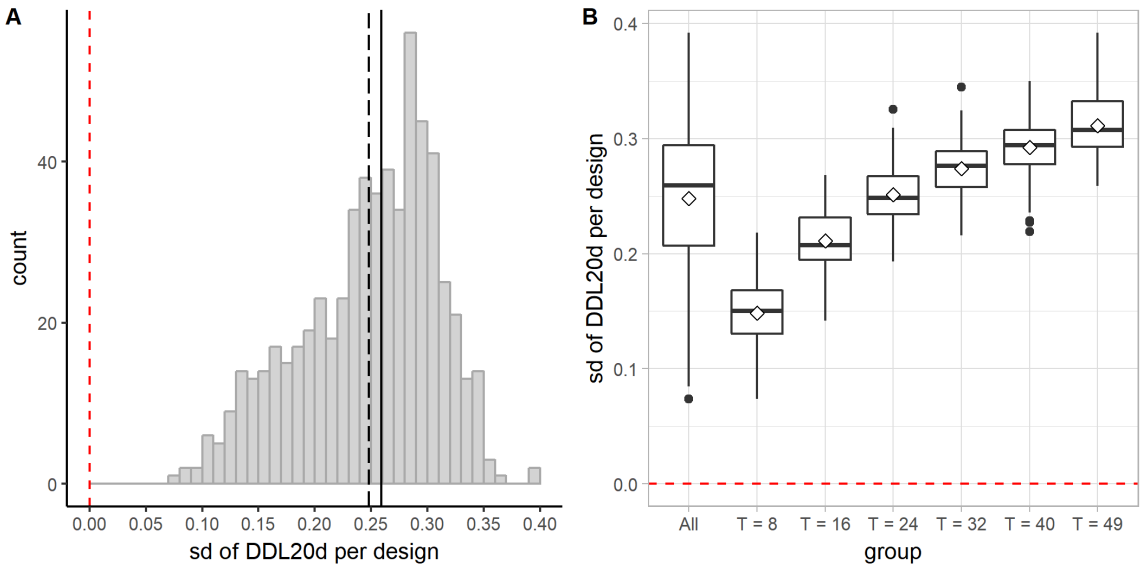

Figure 45. Distribution of the standard deviation of DDL20d per design (shift vertex). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

4.4 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: The operation of shifting vertex, by definition, never changes the topology of a street graph. Therefore, this operation, no matter being applied how many times, has no impact on the topological properties of the street graph, such as the number of vertices, edges, or cells. It has no impact on the vertex degree either. The main effect of this operation is reflected in the change of geometric properties and syntactic conditions in the street graph.

The impact of this operation on the density of a street network: It tends to (a) increase the total street network length per design as well as (b) the mean distance between intersections per design. The first observation follows our intuition: shifting vertices tends to "loosen up" a regular street grid, and the curvilinear streets obviously increase travel distance as compared to the straight-line distance. If we rely on the total street length to evaluate the density of a street network, then this seems to suggest that shifting vertices tends to increase the street network density. However, the observation on the mean distance between intersections somehow challenges this argument. Normally, we would expect a dense street network has not only greater total street length, but also shorter distances between street intersections (Handy, Paterson, & Butler, 2003). In this case, the mean distance between intersections tends to increase too as the total street length increases, which reveals the different nature of the two measures.

The impact of the operation on the directional reach/distance: It tends to increase the mean directional distance between locations. What is more, the effect is attributed to the geometric variation alone. Just by bending/deflecting streets in a design, we can create two designs that differ by one direction change in the mean directional distance per length (20d), which may cause a significant difference in the human perception and

cognition of the environment. The mean directional reach for zero direction change and two direction changes drops as the total street length goes up, indicating the difference between two concepts: the total amount of things nearby, and the accessibility of things conditioned by cognitive distance.

The impact of the operation on the regularity of a street network: It tends to make a street network more irregular. It makes the street network more fragmented (disrupted by sharp turns) as indicated by the measure of "fragmentality". It also tends to create more oddly-shaped blocks, as indicated by the standard area-perimeter ratio. Although it does not change the mean block area per design, it creates more variations in the block area (in other words, the blocks are of more varied sizes).

The impact of the operation on the diversity of syntactic conditions: It tends to increase the diversity of syntactic conditions in a superblock design, as indicated by the increasing number of distinct DDL20d values per design as well as the increasingly greater standard deviations in the DDL20d values per design.

The effects and trends may not always hold as we apply this operation a greater number of times than studied here. The effects discussed above are also dependent on the parameter settings used for generating the sample of designs (e.g., the amount of shift along x- or y-axis).

CHAPTER 5
CONTRACTING EDGE AND CROSS-CONCATENATING VERTICES: CREATING RADIAL STREETS

Part A: Contracting Edge

5.1 Operation: Contract Edge

5.1.1 Definition of operation

The operation of contracting edge collapses an edge {*A*, *B*} into a vertex *C*, and in doing so, it merges the two vertices *A* and *B* into a single vertex (may be called the converging point) *C* which is adjacent to all vertices which were originally adjacent to either *A* or *B* (Figure 46).

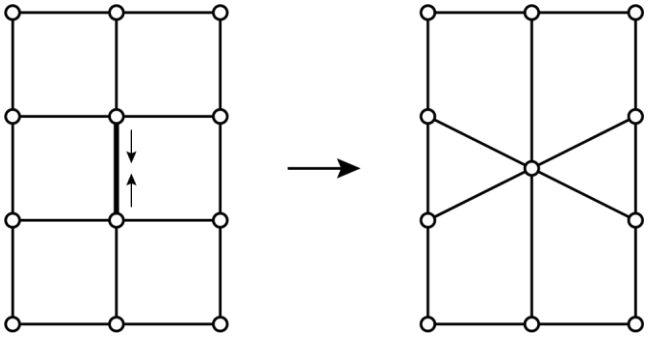

Figure 46. Contract an edge.

5.1.2 Parameters

To precisely control how an edge should be contracted, we need to specify the position of the point to be collapsed to. To specify a relative position along the edge to be

contracted, we can choose one endpoint of the edge, say *A*, to be the reference point and then specify the distance from the point *A* to the point to be collapsed to, say *C*, as shown in Figure 47. Instead of specifying a fixed distance *d*(*A*, *C*), we can specify the ratio between *d*(*A*, *C*) and *d*(*A*, *B*). We can use one of the half-edges associated with the edge {*A*, *B*}, say *h*, to indicate both the edge to be contracted (*h.e*) and the reference point (*h.twin_h.v*).

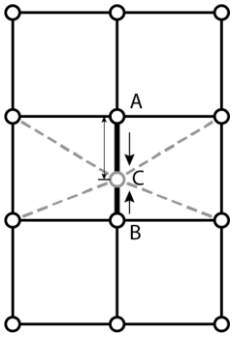

Figure 47. Parametric control of contracting edge.

5.2 Generative Algorithm

5.2.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock designs, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by *X*;

- the number of streets running horizontally in the initial regular grid, denoted by Y;
- the edge to be contracted and the reference point, as indicated by a half-edge h;
- the minimum ratio between the distance from the converging point to the origin vertex of the input half-edge h and the length of the edge $h.e$, denoted by r_{min};
- the maximum ratio between the distance from the converging point to the origin vertex of the input half-edge h and the length of the edge $h.e$, denoted by r_{max}; and
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled via the following parameters: l_x, l_y, X, Y, h, r_{min}, r_{max}, T.

5.2.2 General description

We start from a street graph that represents a regular grid. Then, we randomly pick a half-edge to attempt the operation of edge contraction. We continue to do so until the operation has been successfully performed a specified number of times. The attempted operation can only be successfully performed if *all* of the following conditions are met.

1. The incident edges of the edge represented by the randomly picked half-edge h, as well as the edge itself, do not lie on the border of the street graph.
2. The cell which the half-edge h belongs to does not have any dangling edges or bridges[17] inside.

[17] An edge is a bridge if removing it would make the graph disconnected.

3. The cell which the twin half-edge of *h* belongs to does not have any dangling edge or bridges inside.
4. The next of the next of the next of the half-edge *h* is not *h* itself. (*h.next_h.next_h.next_h* is not *h*)
5. The next of the next of the next of the twin half-edge of *h* is not the twin half-edge of *h* itself. (*h.twin_h.next_h.next_h.next_h* is not *h.twin_h*)
6. After the intended edge-contraction, the angular order of $N(h.v)$ remains the same.
7. After the intended edge-contraction, for each u in $N(h.v)$, the angular order of $N(u)$ remains the same.
8. After the intended edge-contraction, the angular order of $N(h.twin\_h.v)$ remains the same.
9. After the intended edge-contraction, for each u in $N(h.twin\_h.v)$, the angular order of $N(u)$ remains the same.
10. The intended edge-contraction does not lead to any unexpected intersections among edges.

Most of the above conditions are included to simplify the task of maintaining the integrity of the underlying data structure. Below illustrate a few situations where some of the above conditions are not met. Note that the situations illustrated here are hypothetical and may be unlikely or impossible to occur by only applying the operation of contracting edge on a regular grid. Nevertheless, we still check those conditions here so that the proposed algorithm can be used in more general situations.

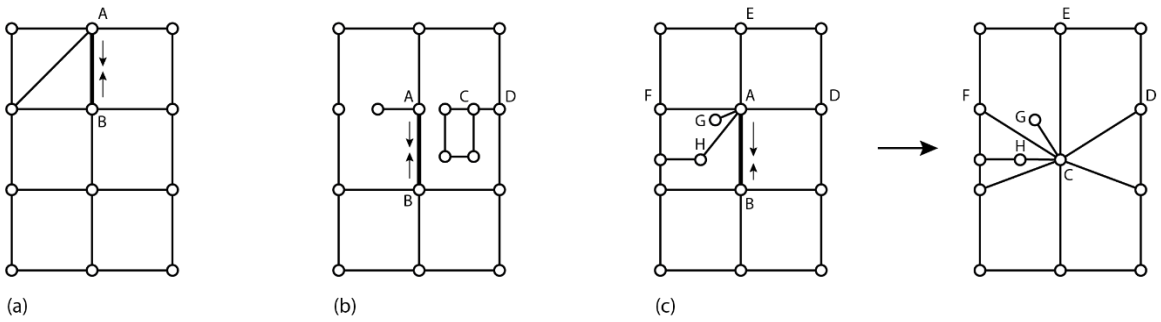

Figure 48. Examples of illegitimate moves (contract edge).

In the example shown in Figure 48a, assume that the half-edge (*B*, *A*) is picked to attempt the contraction of the edge {*A*, *B*}. However, the intended edge-contraction is not allowed because among the edges that are incident with the edge {*A*, *B*}, there are two lying on the border of the street graph—that is, a violation of Condition 1, and also because the next of the next of the next of the half-edge (*B*, *A*) is indeed (*B*, *A*) itself—that is, a violation of Condition 4.

In the example shown in Figure 48b, assume that the half-edge (*B*, *A*) is picked to attempt the contraction of the edge {*A*, *B*}. However, the intended edge-contraction is not allowed because the cell which the half-edge (*B*, *A*) belongs to has a dangling edge—that is, a violation of Condition 2, and also because the cell which the twin half-edge of (*B*, *A*) belongs to has a bridge inside, namely the edge {*C*, *D*}—that is, a violation of Condition 3.

In the example shown in Figure 48c, assume that the half-edge (*B*, *A*) is picked to attempt the contraction of the edge {*A*, *B*}. Before the intended contraction, the angular order (counterclockwise) of the neighborhood of the vertex *A* relative to the ray that starts out at *A* and passes through *B* is *D*, *E*, *F*, *G*, *H*. (We do not consider the vertex *B* itself

here since it will be joined with A after the contraction.) After the intended contraction, the angular order (counterclockwise) of the neighborhood of the vertex *A* becomes *D*, *E*, *G*, *F*, *H*. Since there is a change in order, it violates Condition 6.

5.2.3 Pseudocode

Suppose that we have already generated the initial street graph *G* which represents a regular grid that is l_x units long, l_y units wide, with *X* number of streets running vertically and *Y* number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of contracting edge based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random* (*a*, *b*) refers to a procedure which returns a random floating-point number *N* such that $a \leqslant N \leqslant b$ for $a \leqslant b$, and *random_choice* (*seq*) refers to a procedure which returns a random element from the sequence *seq*.

ALGORITHM 5.1: Contract Edges

procedure *contract_edges* (*G*, r_{min}, r_{max}, *T*)
counter ≔ 0
while *counter* < *T*
    h ≔ a randomly picked half-edge from the half-edge set *H* of *G*
    if *h.e* and the edges incident with *h.e* do not lie on the border of *G*
    and *h.c* does not have any dangling edges or bridges inside
    and *h.twin_h.c* does not have any dangling edges or bridges inside
    and *h.next_h.next_h.next_h* is not *h*
    and *h.twin_h.next_h.next_h.next_h* is not *h.twin_h*
    and after the intended edge-contraction,
    the angular order of *N*(*h.v*) remains the same
    and for each *u* in *N*(*h.v*), the angular order of *N*(*u*) remains the same
    and the angular order of *N*(*h.twin_h.v*) remains the same
    and for each *u* in *N*(*h.twin_h.v*), the angular order of *N*(*u*) remains the same
    and it does not lead to any unexpected intersections among edges **then**
        r ≔ *random* (r_{min}, r_{max})
        p ≔ (*h.twin_h.v.x* + *r* (*h.v.x* – *h.twin_h.v.x*), *h.twin_h.v.y* + *r* (*h.v.y* – *h.twin_h.v.y*))
        contract *h.e* to the point *p*
        counter ≔ *counter* + 1

5.3 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the six groups of designs, we varied only one control parameter—namely *T*, the total number of operations to be performed—but kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9, r_{min} = 0.5, r_{max} = 0.5. By assigning the same value (0.5) to both r_{min} and r_{max}, we specify that the edge should always be collapsed to its midpoint after contraction.

Although the effect of a single application of the operation seems to be clear, the effect of multiple applications of the operation is much less so. As shown in Figure 49, contracting the edge {*A*, *B*} deflects the two horizontal streets originally connected by it.

The angle of deviation as one travels from the edge {*E, F*} to {*F, G*} or from the edge {*E, H*} to {*H, J*}, marked by α, is about 27°. Assume that the threshold angle used to indicate a direction change is 20°, then contracting the edge {*A, B*} creates additional turns on the otherwise straight streets. However, it would be wrong if we conclude that contracting an edge always deflect streets and thus create additional turns. For instance, if we contract the edge {*C, D*} following the contraction of {*A, B*}, the angle of deviation it causes as one travels from the edge {*M, E*} to {*E, K*}, marked by β, is only about 18°. Since it is less than the threshold angle (20°), it creates a longer linear extension for the street passing the edge {*M, E*} instead of adding additional turns to it.

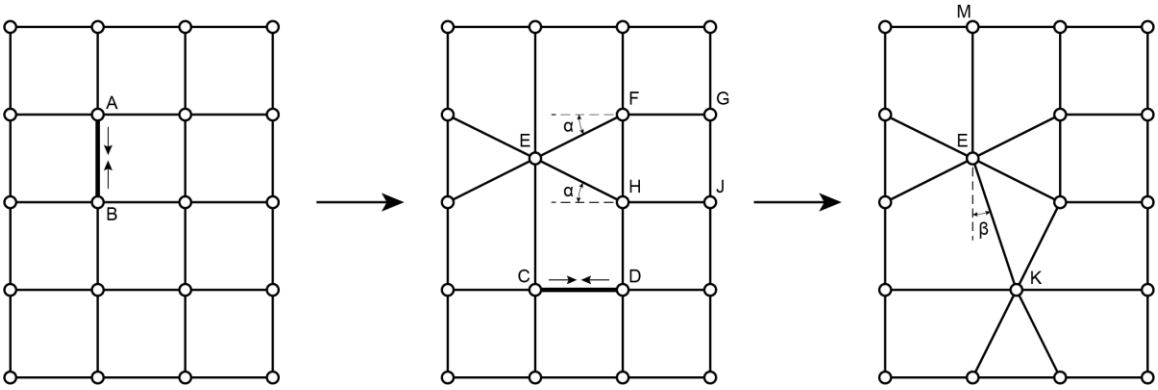

Figure 49. Angle of deviation caused by the operation of contracting edge.

5.3.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation (i.e., the appropriate value of *T*) on the initial 9 × 9 square-grid design. We adopted the following strategy: First, we choose an arbitrarily high number *a* so that it is impossible to apply the operation of cross-concatenating vertices more than *a* times on the initial square-grid design based on our

algorithm. We run the algorithm by setting $T = a$ and a terminating condition: If it has made 2000 attempts in a row and none of them are successful, then it will stop attempting the operation anymore and record the total number of times that the operation has been successfully applied, denoted by b. This gives us a sense of the upper limit for the number of successful applications of the operation for a particular run. However, the recorded number b may vary for different runs. Thus, we make 100 trial runs and record the number b for each trial. Then we choose the minimum recorded value for b among all the trial runs as the maximum number of times we would apply this operation to generate a group of designs, denoted by T_{max}. The minimum number of times that we would apply the operation, T_{min}, is subsequently determined by finding the nearest integer for $T_{max} / 6$. To generate the other four groups of designs, we simply set T to $2T_{min}$, $3T_{min}$, $4T_{min}$, $5T_{min}$, respectively.

Examples of designs from each group are shown in Figure 50. Note that the blocks (cells) consist of only quads and triangles.

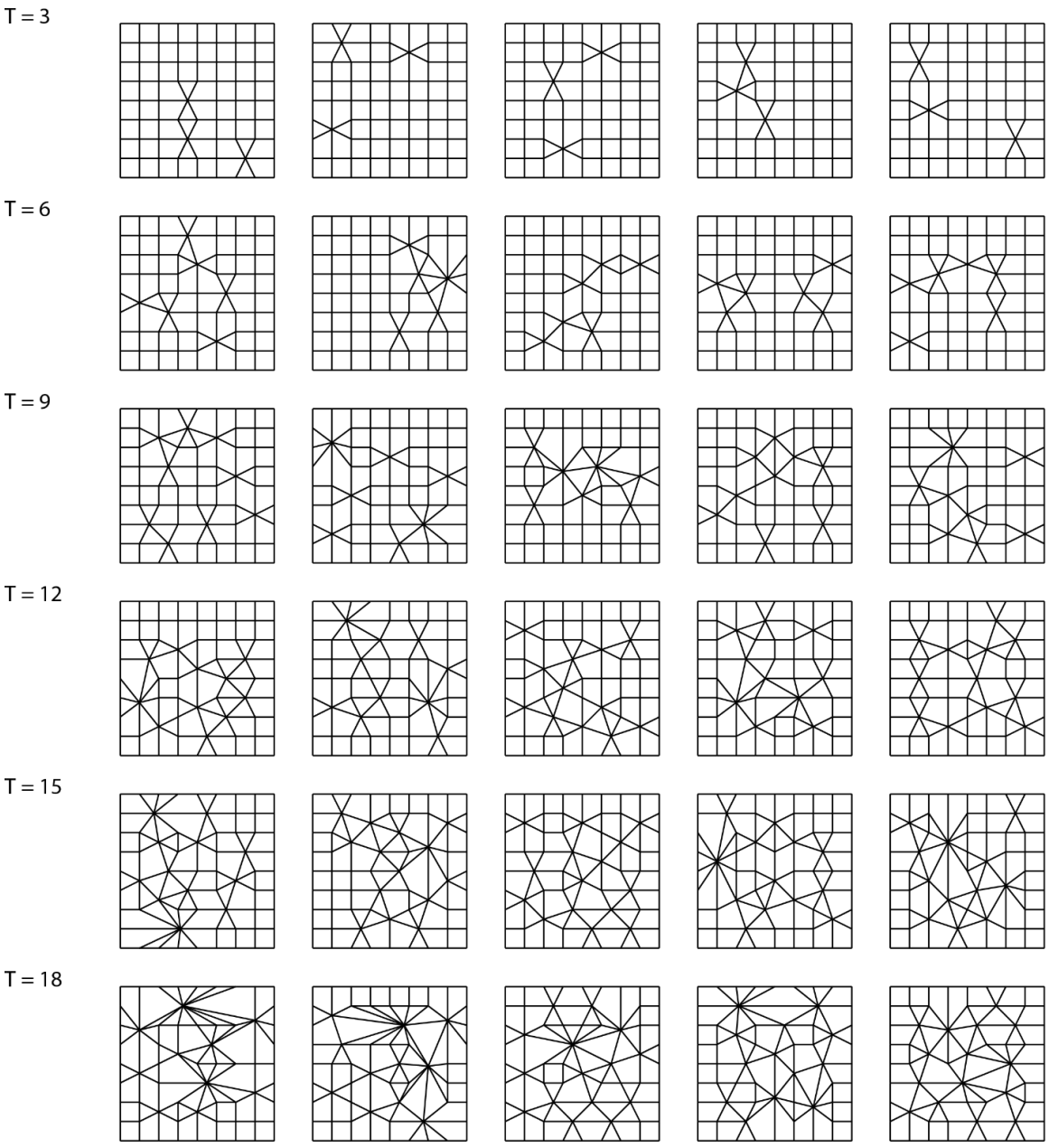

Figure 50. Examples of designs from each group (contract edge).

5.3.2 Data analysis

Six groups of designs were generated by applying the operation of contracting edge 3, 6, 9, 12, 15, and 18 times, respectively. Each group consists of 100 designs. The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of designs.

Elementary graph properties

(1) Number of vertices

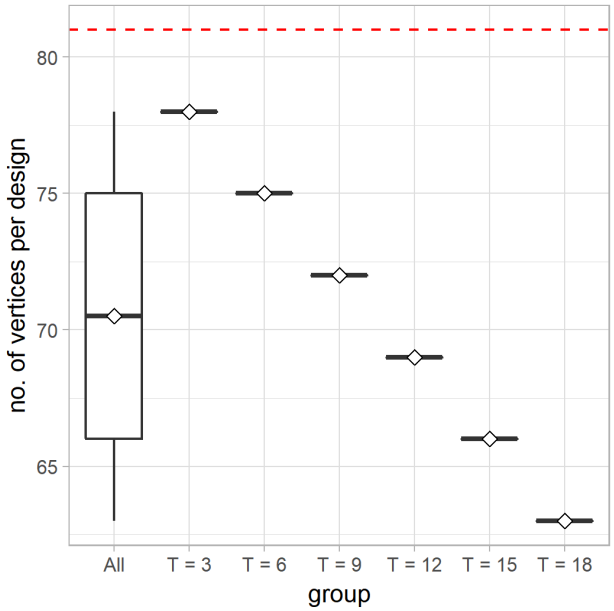

Figure 51. A boxplot (with mean diamonds) showing the distribution of the total number of vertices per design for all designs and for each group of designs (contract edge). The red dashed line indicates the total number of vertices in the initial 9 × 9 square-grid design (81).

When an edge is contracted, the two vertices at its two endpoints are merged, thus reducing the total number of vertices in the street graph by one. As shown in Figure 51,

the total number of vertices per design drops steadily as more edges are contracted during the generative process. For a design generated by contracting T edges in the initial street graph, we know that $|V| = 81 - T$.

(2) Number of edges

The conditions we imposed above ensure that contracting an edge only reduces the total number of edges by one. As shown in Figure 52, the total number of edges per design decreases as more edges are contracted during the generative process. For a design generated by contracting T edges in the initial street graph, we know that $|E| = 144 - T$.

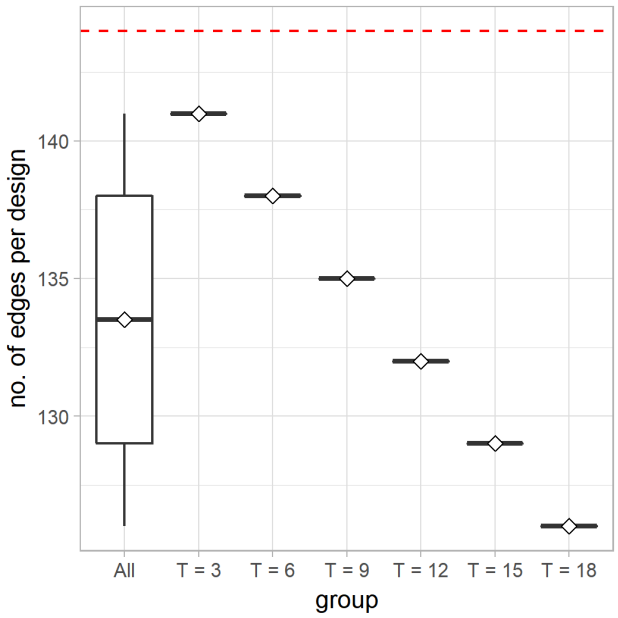

Figure 52. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (contract edge). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

The conditions we imposed above ensure that contracting an edge does not change the total number of cells in a design. Therefore, no matter how many times we have applied the operation, we know that $|C| = 64$.

(4) Vertex degree

As shown in Figure 53, most vertices have degree less than or equal to 4. The most common vertex degree for all the vertices analyzed is 4, with the second most common degree being 3. Nevertheless, the maximum degree observed for a vertex is 16—that is, four times as large as the most common degree for a vertex. Higher vertex degrees appear in groups of designs generated by more frequent contraction of edges. It is not surprising at all: when an edge is contracted, the edges originally incident with its two endpoints are now joined to a single vertex, thus contributing to a higher vertex degree than the degree of either endpoint of the original edge. In addition, since initially all the vertices inside the boundary of the street graph have degree 4 and, based on our algorithm, the edges connecting the vertices that have degree 3 (i.e., the vertices lying on the boundary of the street graph) are not allowed to be contracted, the vertex degree is always incremented by an even number when an edge is contracted. This also explains why the outliers are all located at even values in the boxplots shown in Figure 53B. The mean vertex degree per design is higher for designs generated by more frequent contraction of edges. Moreover, the mean vertex degree for the designs generated by applying the operation T times on the initial 9×9 square-grid design equals $2(144 - T) / (81 - T)$.

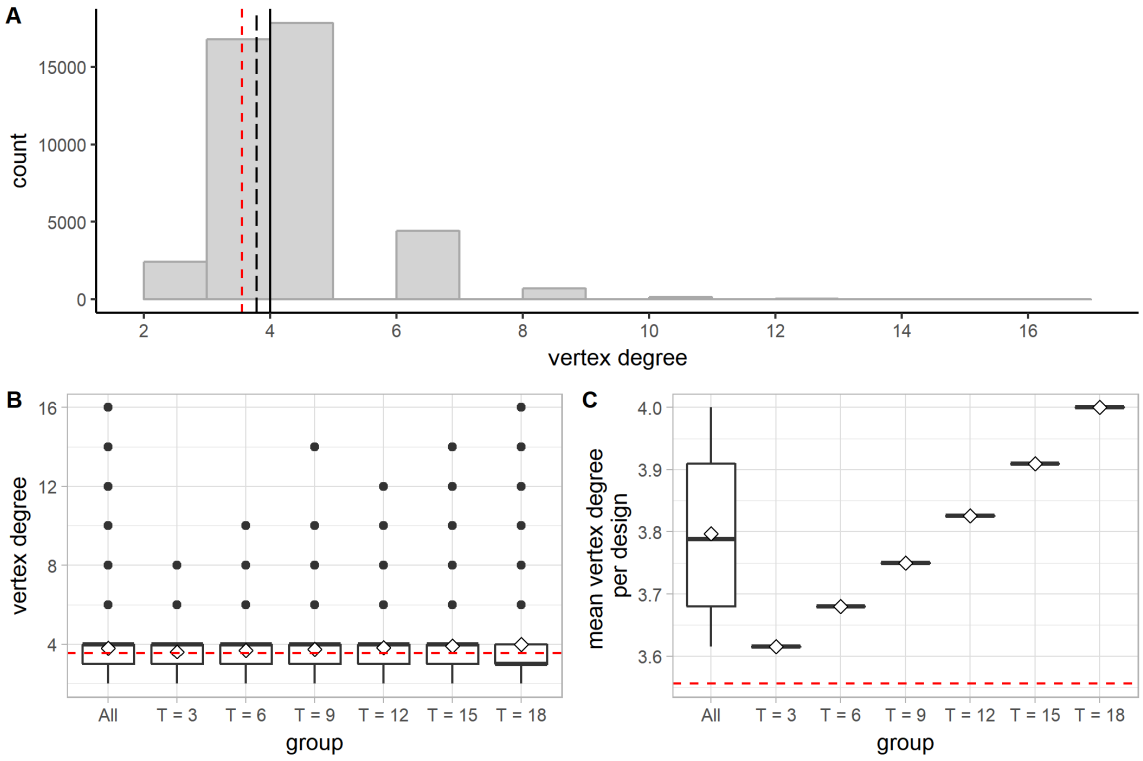

Figure 53. Distribution of the vertex degree (contract edge).A: A histogram showing the distribution of the vertex degree for all vertices in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design (≈ 3.556). B: A boxplot (with mean diamonds) showing the distribution of the vertex degree for all vertices and for vertices in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

As shown in Figure 54, the total street length increases as edges are contracted in the initial 9 × 9 square-grid design. While almost half of all designs have a total street length of between 14500 and 15000 m, the maximum total street length per design could go above 16700 m, an over 15% increase in the street length compared to the initial

square-grid design. Moreover, the total street length per design tends to increase as edges are contracted more frequently in the generative process. The groups of designs generated by more frequent application of this operation also tend to cover a wider range of values, mainly because they include designs that score extremely high based on total street length.

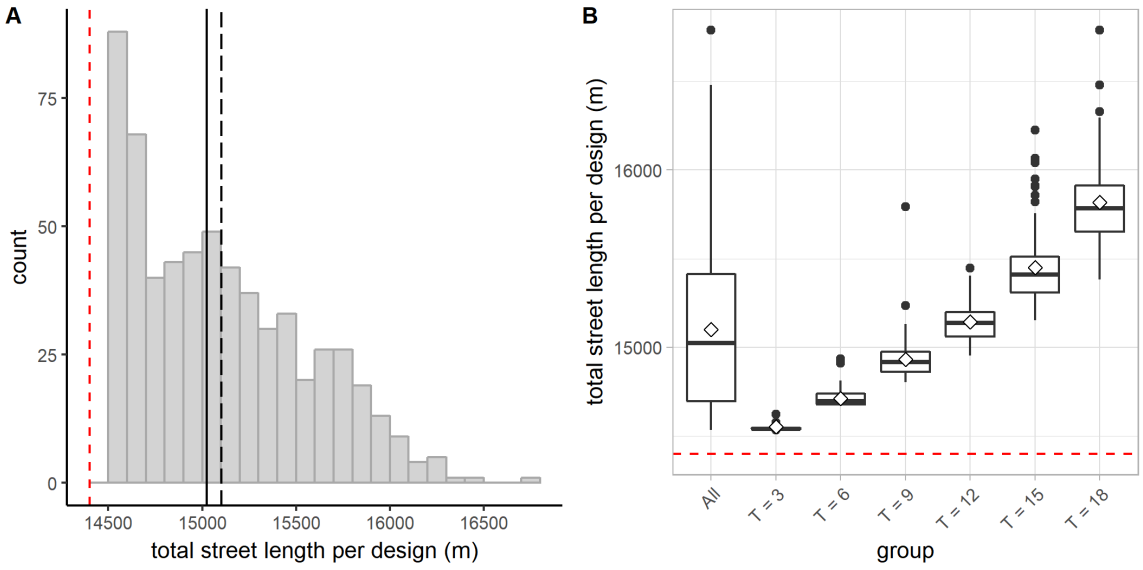

Figure 54. Distribution of the total street length per design (contract edge). A: A histogram showing the distribution of the total street length per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs.

(2) Total number of blocks per design

Since contracting an edge does not change the total number of cells in a design, the total number of blocks in each generated superblock design is 64, the same as in the initial 9 × 9 square-grid design.

(3) Total number of intersections per design

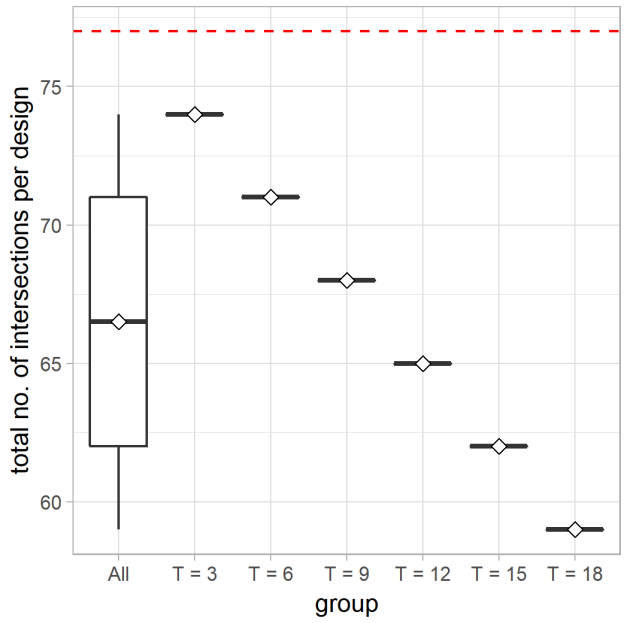

Figure 55. A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs (contract edge). The red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77).

The total number of intersections in a design is equal to the total number of vertices that have degree greater than or equal to 3 in the design. Since the operation of contracting edge, by definition, never decreases vertex degree, the change in the number of intersections is the same as the change in the number of vertices. More specifically, the total number of intersections per design equals $|V| - 4$. We subtract 4 from $|V|$ because the

four vertices located at the corners of the superblock have degree 2 and cannot be counted as intersections. The total number of intersections per design decreases as more edges are contracted (Figure 55). For designs generated by contracting T edges based on Algorithm 4.1, the total number of intersections per design is $(81 - T) - 4$, or $77 - T$.

(4) Distance between intersections

As shown in Figure 56, the distance between intersections is never below 100 m (which is the distance between intersections in the initial square-grid design). In other words, applying the operation of contracting edge on a square grid seems to only increase the distance between intersections. The distance between intersections is never shorter than 100 m and it can exceed 450 m. Longer distances between intersections are observed more often in designs generated by more frequent contraction of edges (Figure 56B). The mean distance between intersections per design also tends to increase as more edges are contracted in the generative process (Figure 56C). From the point of view of the mean distance between intersections, the best design among the group of designs with $T = 18$ is still worse than the worst designs in the groups of designs with $T = 3$, 6, 9, and 12.

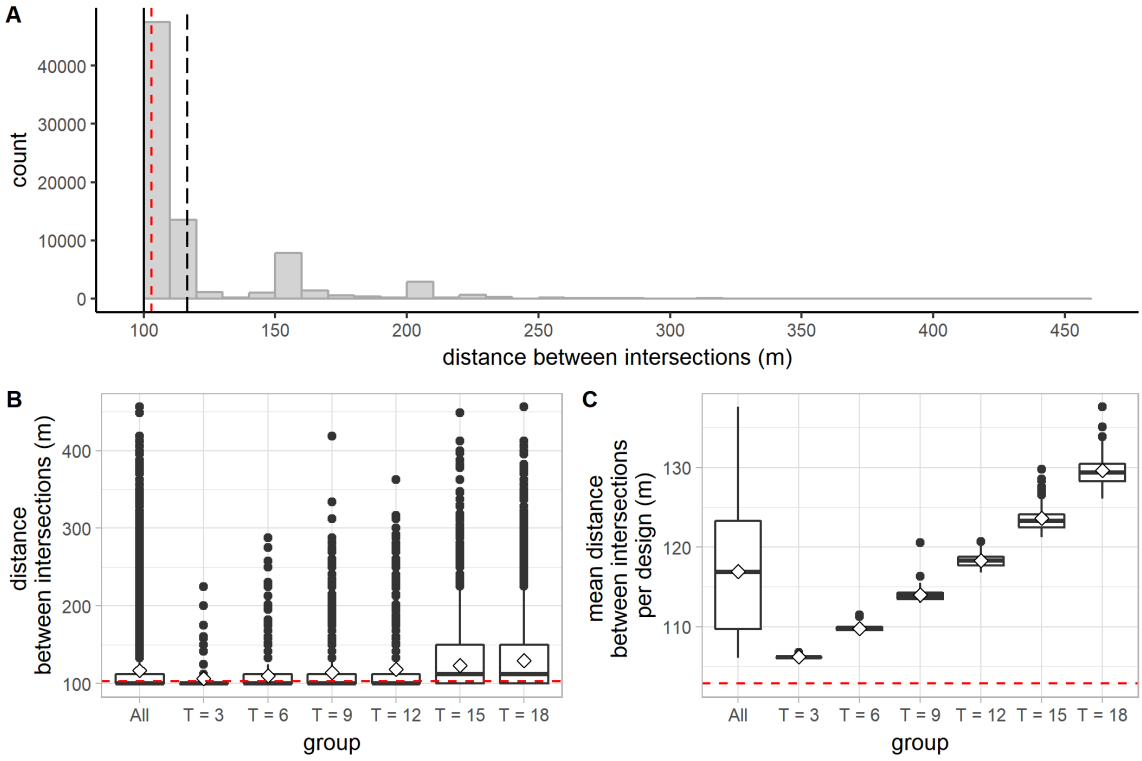

Figure 56. Distribution of the distance between intersections (contract edge).. A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the distance between intersections observed in all designs and in each group of designs. C: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Directional reach and directional distance

(1) DDL (directional distance per length)

As shown in Figure 57, the distribution of the DDL values is dependent on how we set the threshold angle. The mean DDL drops nearly 30% as the threshold angle is increased from 10° to 30°. In the following analysis, unless otherwise specified, the directional distance is always evaluated by setting the threshold angle to 20°.

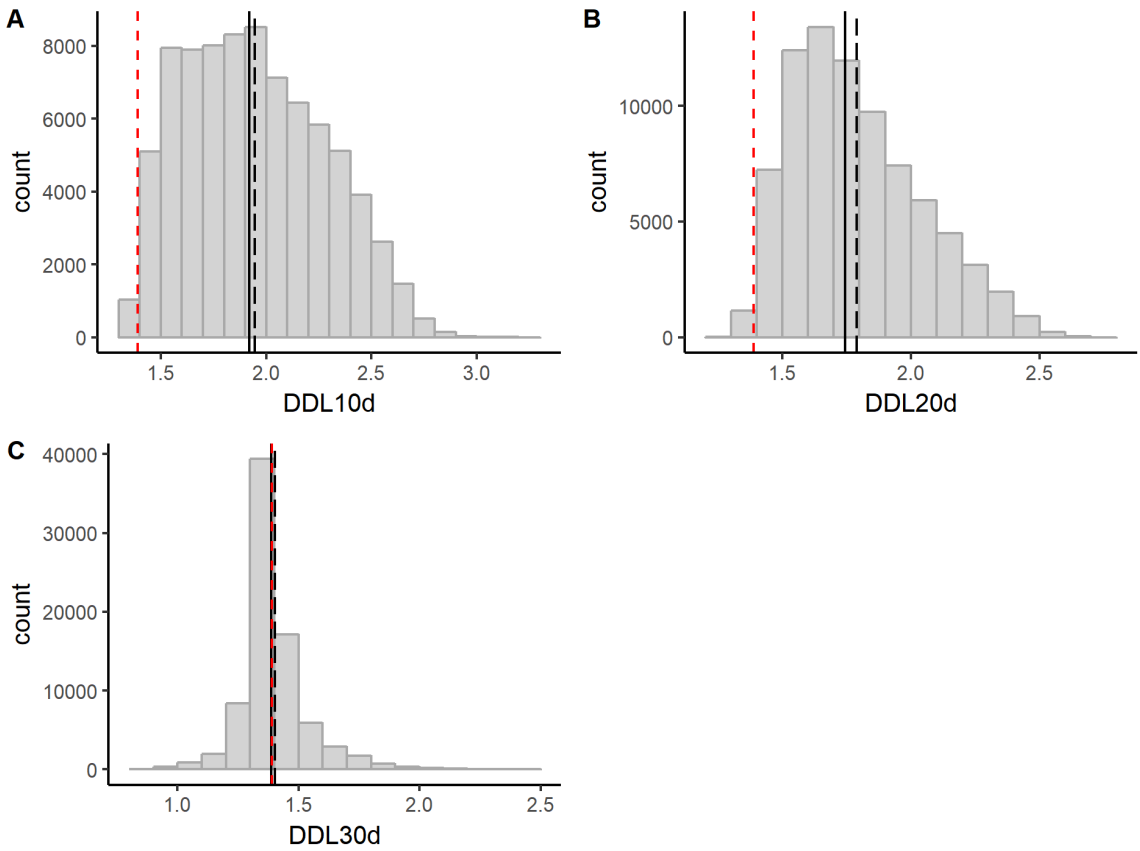

Figure 57. Distributions of DDL10d, DDL20d, and DDL30d (contract edge). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

As shown in Figure 57B and Figure 58A, the distribution for DDL20d is skewed to the right. The values of DDL20d for all segments range from 1.23 to 2.71. More than half of all segments analyzed assume a DDL20d value lower than 1.8. The segments in the groups of designs generated by more frequent contraction of edges assume a wider range of values. More importantly, more frequent contraction of edges could create not only more segregated segments but more integrated (i.e., lower DDL20d) ones as well (Figure 58A). Initially, the mean DDL20d per design tends to increase as more edges are contracted, however, the effect becomes obscure after a certain number of applications of

the operation. As shown in Figure 58B, there is a rising trend in the mean DDL20d per design when the number of edges contracted increases from 3 to 12. However, the trend disappears when the number of edges contracted is greater than 12. The mean of the mean DDL20d per design for the group with $T = 12$ is similar to (even slightly higher than) those for the groups with $T = 15$ and $T = 18$ (Figure 58B).

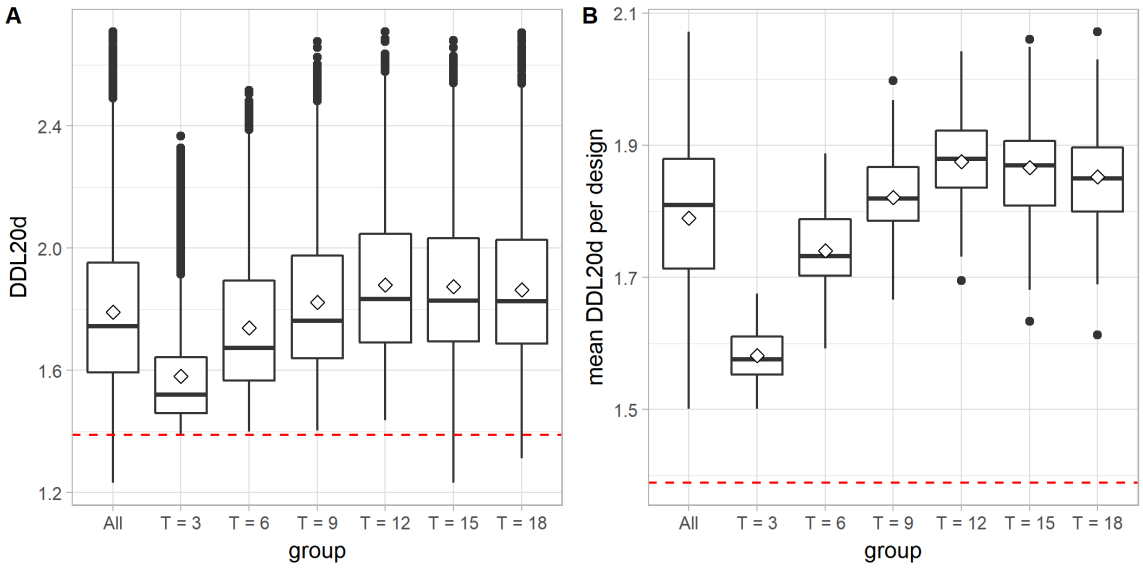

Figure 58. Distribution of DDL20d (contract edge).A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 59, almost half the segments have a linear reach of more than 600 m, and nearly 45% of all the segments have a linear reach of between 800 and 850 m—that is, about the same as the width of the entire superblock. The linear reach for all segments covers a wide range of values, from 100 m to 2200 m. (Extremely long linear

reach can be achieved when multiple branching pathways align relatively linearly with the segment under study.)

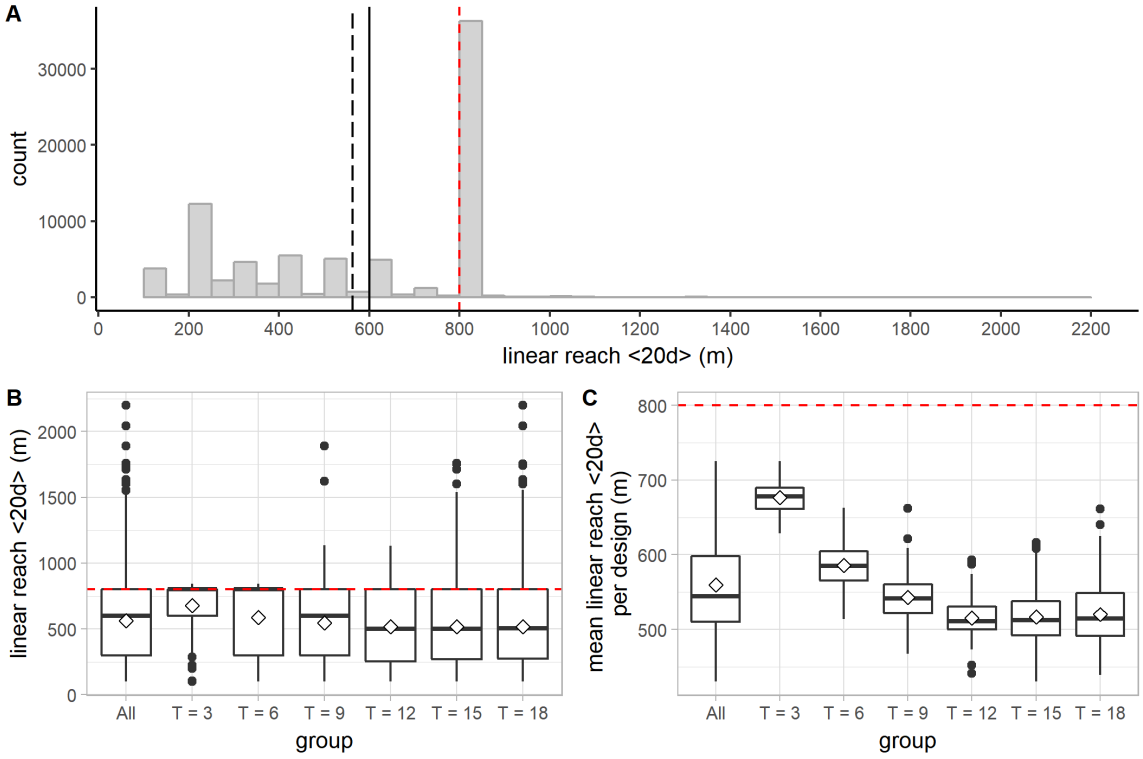

Figure 59. Distribution of the linear reach (contract edge). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

When analyzed in groups, the mean linear reach decreases as the number of edges contracted increases from 3 to 12. Contracting more edges after that point, however, seems to have no clear effect on the mean linear reach when analyzed by group (Figure 59B). On the other hand, increasingly higher linear reach values appear as more edges are contracted, widening the range of the values (although the interquartile range changes

very little). Likewise, initially, the mean linear reach per design tends to decrease as the edges are contracted more frequently. However, the effect becomes unclear after the operation has been applied more than 12 times (Figure 59C).

(3) 2-dc reach (dr2dc20d)

As shown in Figure 60, The values of 2-dc reach range from about 5400 m to nearly 17000 m. The distribution of 2-dc reach for all segments is heavily left-skewed. Almost half of all segments have a 2-dc reach of greater than 13500 m.

When analyzed in groups, the range of 2-dc reach values generally becomes wider for the segments in the groups of designs generated by more frequent contraction of edges. As edges are contracted more often, extreme 2-dc reach values tend to appear. The effect of the operation on the measure of 2-dc reach is not straightforward: although the mean 2-dc reach for the segments grouped by T drops when T increases from 3 to 12, it slightly rises when T is increased from 12 to 18 (Figure 60B). Likewise, the mean 2-dc reach per design tends to cover a wider range of values as edges are contracted more frequently during the generative process. The mean of the mean 2-dc reach per design, when grouped by T, drops when T is increased from 3 to 12, then rises when T is increased from 12 to 18 (Figure 60C). Note that there are designs in the group with $T = 3$ scoring higher than the initial 9 × 9 square-grid design in terms of mean 2-dc reach.

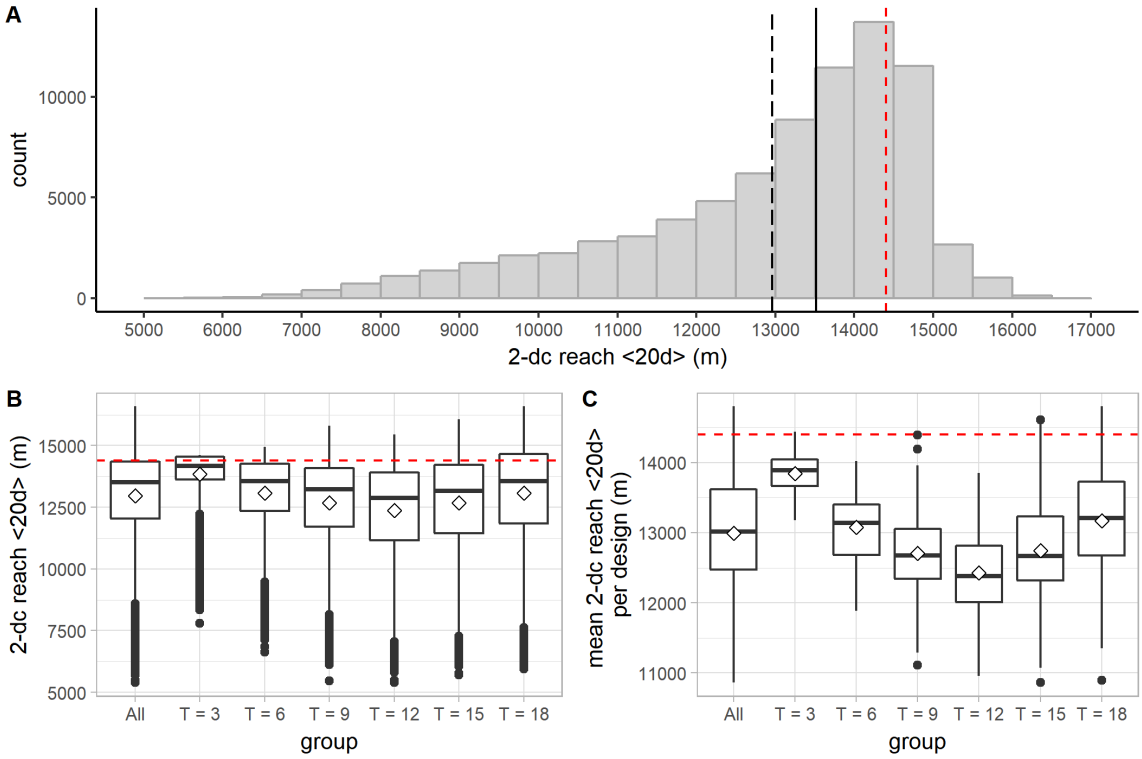

Figure 60. Distribution of the 2-dc reach (contract edge). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

As shown in Figure 61, the fragmentality per design ranges from 0.16 to 0.44. Half of all designs have a fragmentality value greater than 0.3. The fragmentality per design tends to increase as more edges are contracted. Note that, from the perspective of

fragmentality, the most deformed design in the group with $T = 3$ is still more regular than the most regular design among all the other groups of designs (Figure 61B).

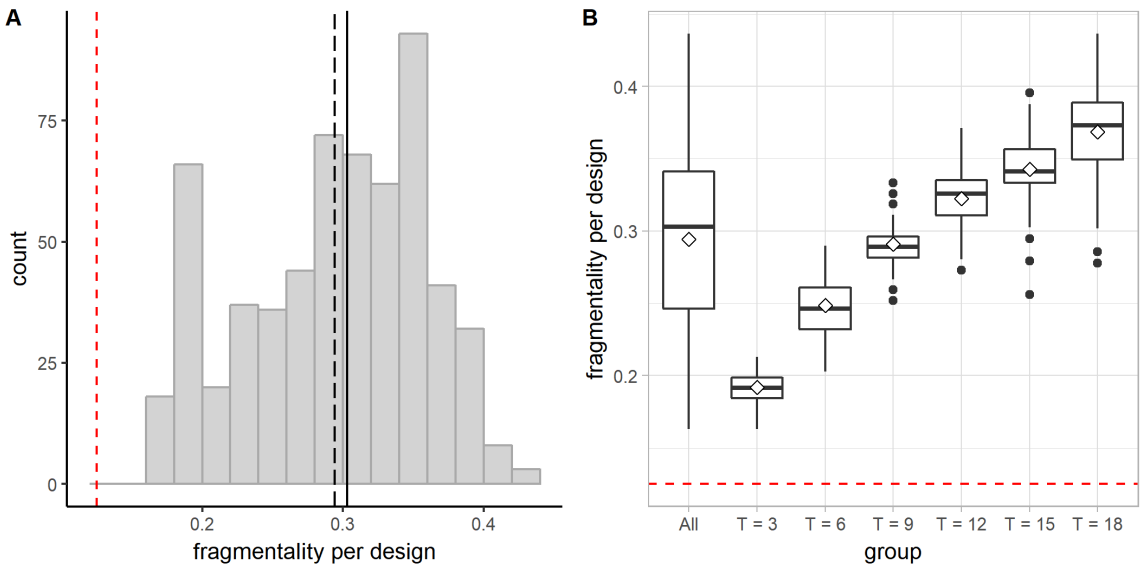

Figure 61. Distribution of the fragmentality per design (contract edge). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

As shown in Figure 62, the area of a block ranges from below 200 m² to more than 50000 m². In other words, it could be smaller than 1/50 of a 100 × 100 m block, or more than four times larger than a 100 × 100 m block. More than 65% of all blocks have an area smaller than or equal to 10000 m². There is a notable peak for the bin located at around 5000 m², suggesting the prevalence of triangular blocks resulted from edge-contraction whose size is half that of the original 100 × 100 m block. When grouped by *T*, a wider range of values are observed for the groups generated by more frequent edge

contractions. However, since the number of blocks stays the same during the generative process, the mean block area per design stays the same as the initial 9 × 9 square-grid design.

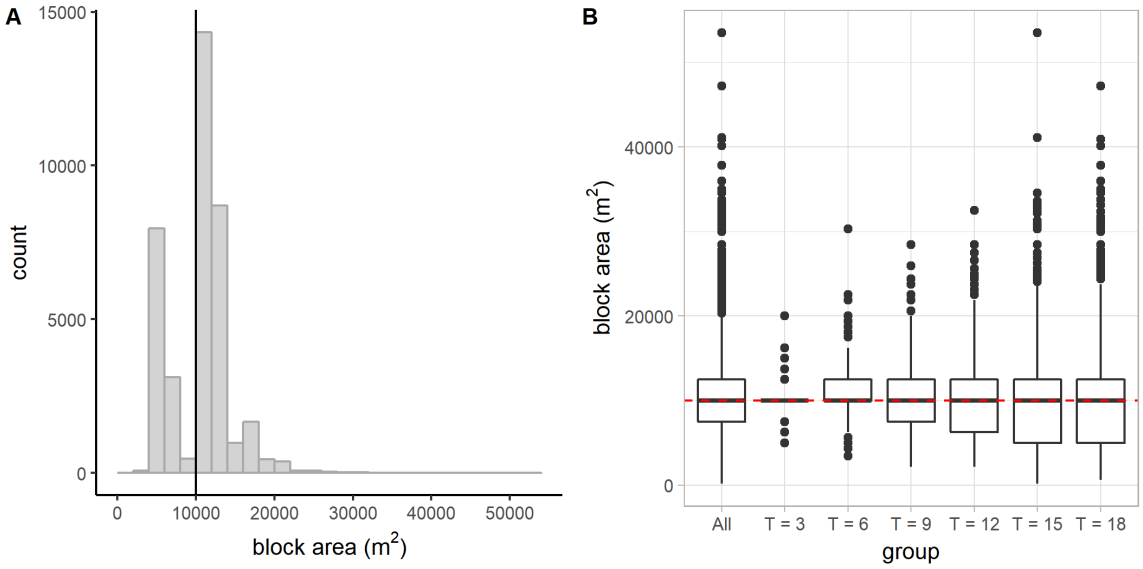

Figure 62. Distribution of the block area (contract edge). A: A histogram showing the distribution of the area for all the blocks in all designs. The solid black line indicates the median. B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs. The red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²).

While the mean block area per design stays the same throughout the generative process, the areas of the blocks become more and more varied as the operation is applied more frequently, as evidenced by the increasing standard deviation of the block area per design (Figure 63).

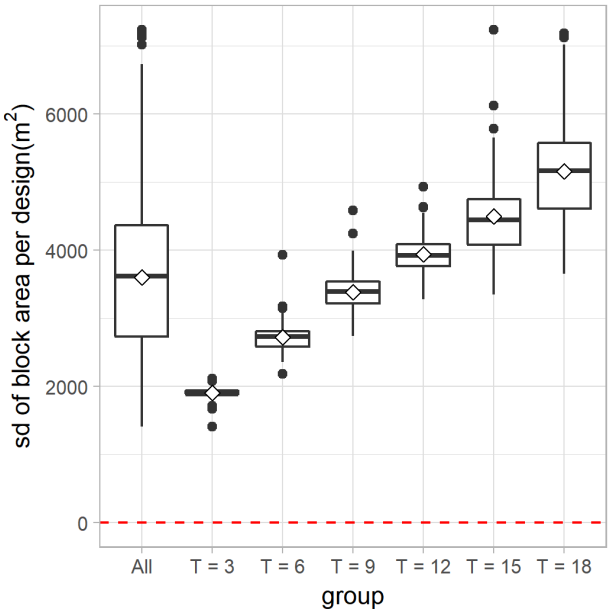

Figure 63. A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs (contract edge). The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²).

(3) Block perimeter

As shown in Figure 64, the perimeter of a block ranges from below 350 m to over 1000 m. Nearly 60% of all blocks have a perimeter of less than or equal to 400 m. Thirty-five percent of all blocks have a perimeter of exactly 400 m. As shown in Figure 64B, the blocks with the greatest perimeter length belong to the groups of designs generated by more frequent edge-contraction. The block perimeter also varies over a wider range of values for those groups of designs. The mean block perimeter per design tends to increase for the designs generated by more frequent contraction of edges (Figure 64C). The distribution is identical to the total length per design, because the mean block perimeter can be roughly computed by dividing the total street length by the total number of blocks

in a design. Since the total number of blocks per design stays the same for all designs analyzed here, the mean block perimeter is only influenced by the total street length.

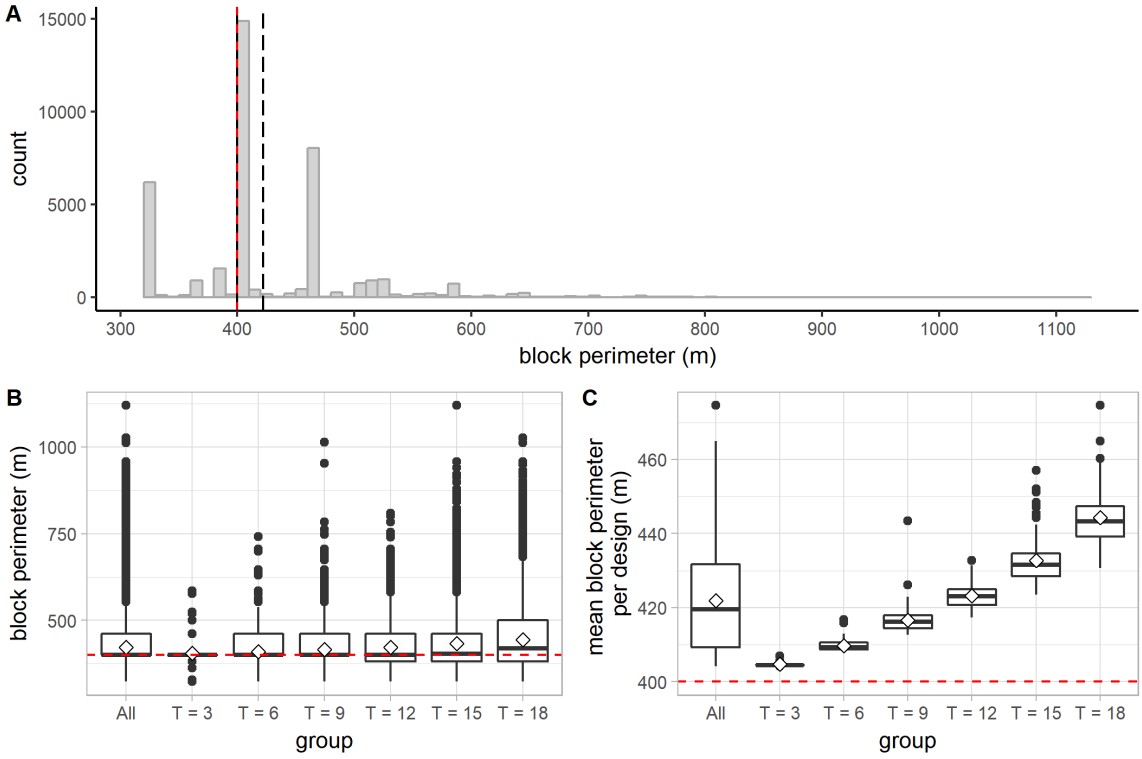

Figure 64. Distribution of the block perimeter (contract edge). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

The perimeters of blocks inside a design become more and more varied as the operation is applied more and more frequently, as evidenced by the increasing standard deviation of the block perimeter per design (Figure 65).

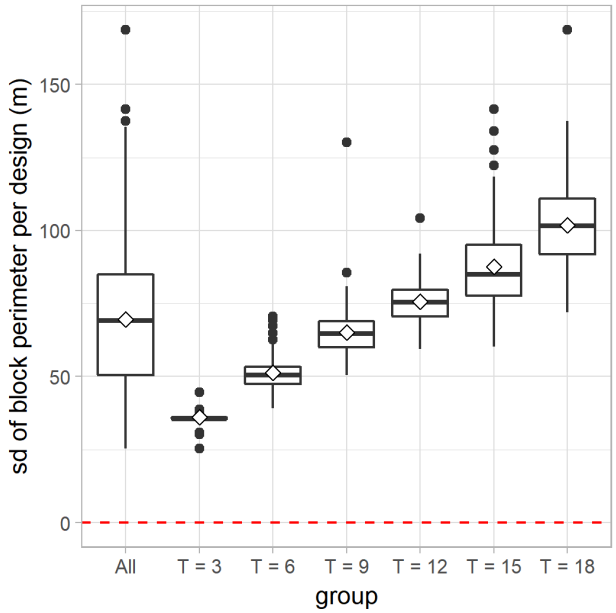

Figure 65. A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs (contract edge). The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m).

(4) Standardized block area-perimeter ratio (SAPR)

As shown in Figure 66, the standardized block area-perimeter ratio (SAPR) for all blocks ranges from below 0.1 to 1.0. Almost 60% of all blocks have a SAPR greater than 0.96, or closely resemble a perfect square in terms of area-perimeter ratio. When analyzed in groups, the SAPR tends to drop as more edges are contracted. In other words, we are more likely to produce oddly shaped blocks as we contract more edges in a design—for instance, the lowest SAPR we get from the group of designs with $T = 15$ is only 0.08, while the lowest SAPR from the group with $T = 3$ is still above 0.75 (Figure 66B). The mean SAPR per design tends to decrease for designs generated by more frequent contraction of edges (Figure 66C).

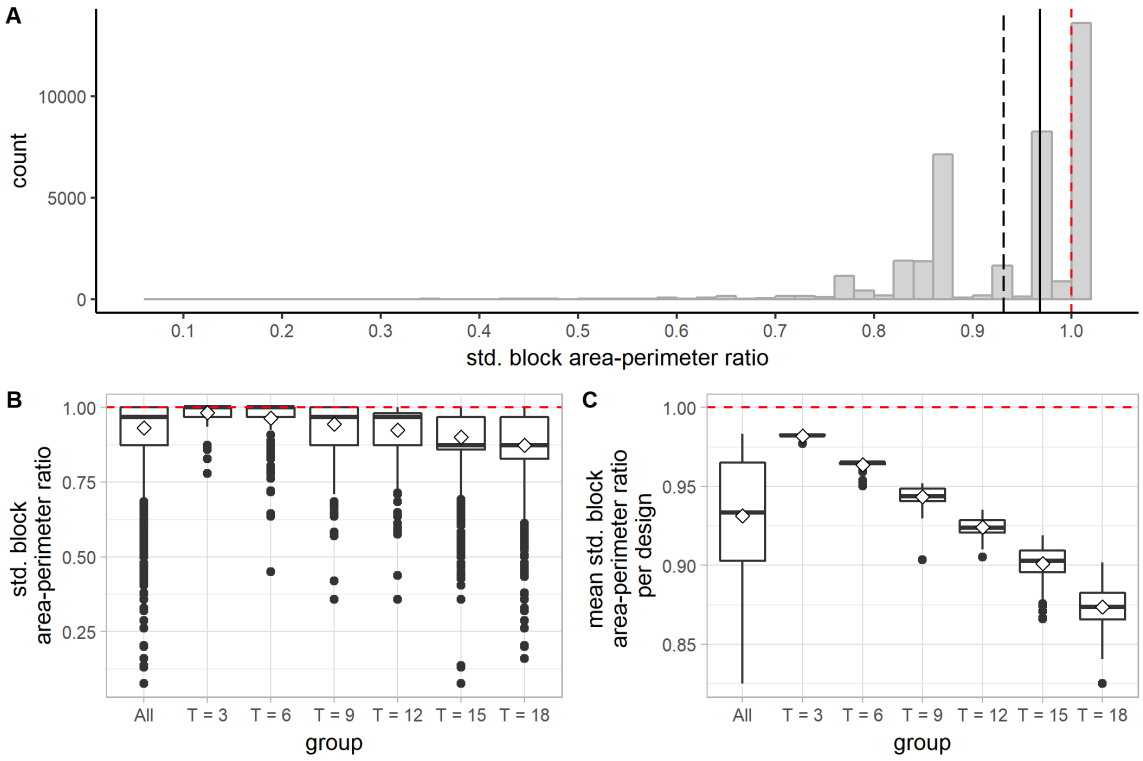

Figure 66. Distribution of the standardized block area-perimeter ratio (SAPR) (contract edge).A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and proportion of distinct DDL20d values per design

As shown in Figure 67, the total number of distinct DDL20d values per design ranges from 11 to 60. About half designs have 40 or more distinct DDL20d values. Although the total number of edges per design decreases when more edges are contracted during the generative process, the total number of distinct DDL20d values tend to increase in designs generated by more frequent contraction of edges (Figure 67B). To

make a fair comparison, instead of directly comparing the absolute numbers, we further divide the total number of distinct DDL20d values by the total number of segments in the design. It is clear that the proportion of unique DDL20d values per design also tends to increase for the designs generated by more frequent contraction of edges (Figure 67C).

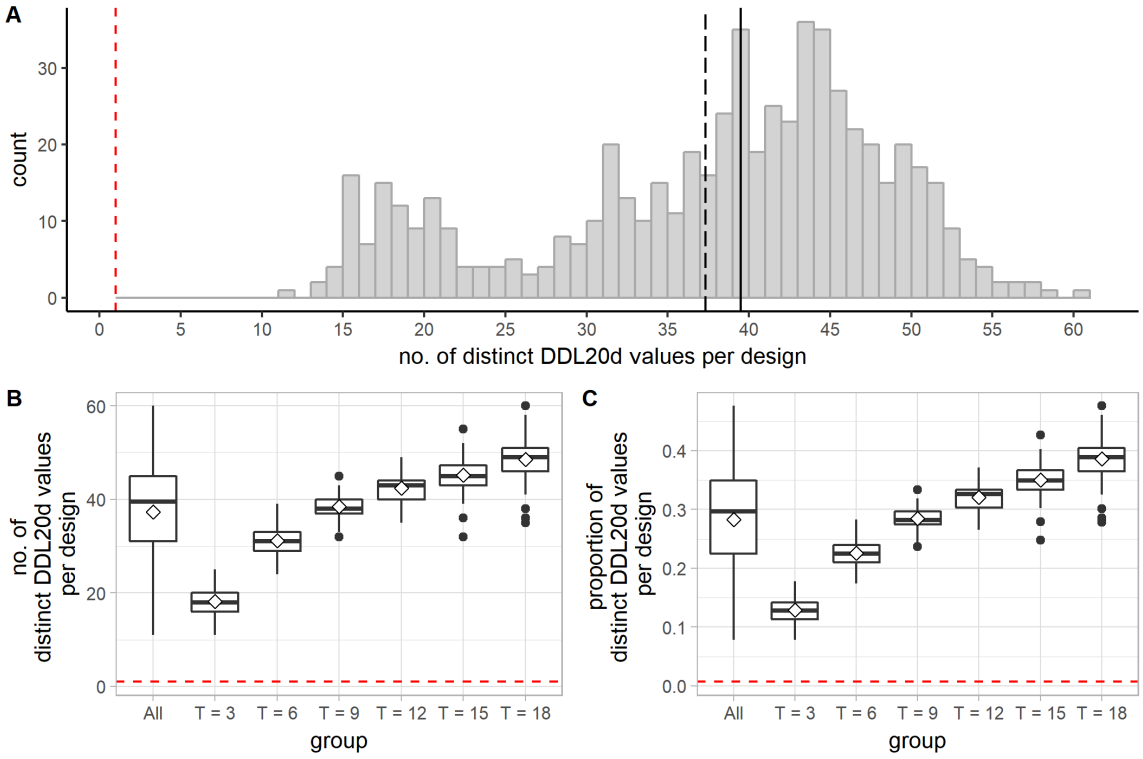

Figure 67. Distribution of the total number and the proportion of distinct DDL20d values per design (contract edge). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d per design

As shown in Figure 68, the standard deviation of DDL20d values per design ranges from about 0.11 to about 0.28. It tends to increase as edges are contracted more frequently. However, after a certain number of edges have been contracted, the standard deviation of DDL20d tends to stabilize, as shown in Figure 68B.

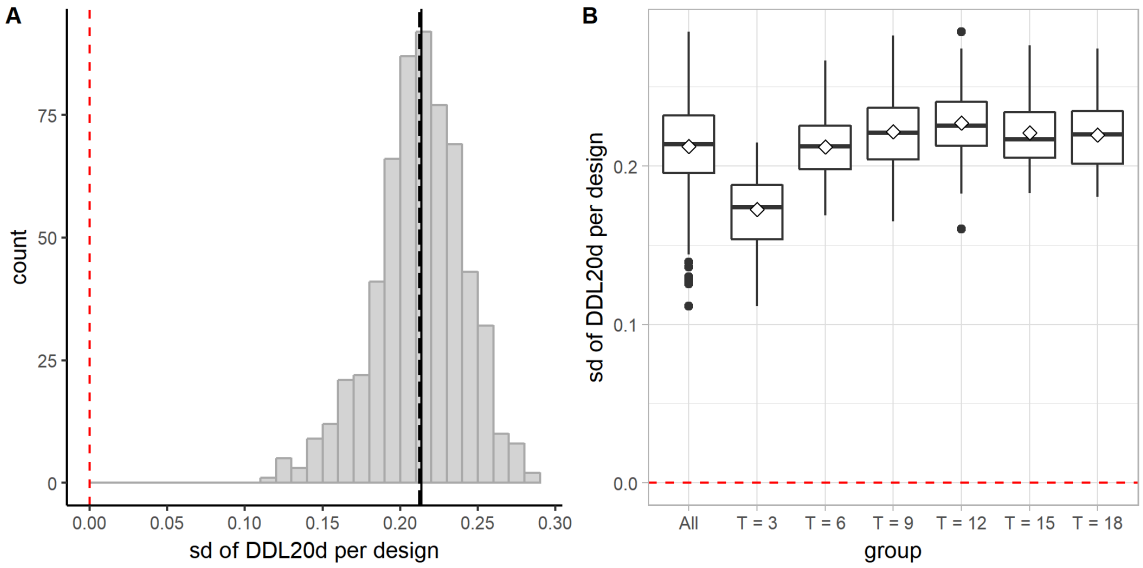

Figure 68. Distribution of the standard deviation of DDL20d per design (contract edge). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

5.4 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: based on our algorithm, contracting an edge would reduce the total number of vertices by one, reduce the total number of edges by one, and maintain the total number of cells. Contracting an edge, according to the rules we specified, always creates a vertex equal to or higher than the degree of either endpoint of the original edge. Thus, multiple application of this operation tends to produce vertices of higher degrees and increase the mean vertex degree per design.

The impact of the operation on the density of a street network: This operation tends to increase the total length of streets in a superblock design. In the extreme case, the total street length could be increased by more than 15% by applying this operation alone on the initial square-grid design. It may seem a bit counterintuitive at first glance: why the total length of streets is increased while the total number of edges is reduced? Partially it is because the rules we specified ensures that only one edge is lost during each edge-contraction (thus situations where multiple edges be eliminated during one edge-contraction will not happen based on our algorithm); it is also because although one edge is lost by contraction, the length of its loss are often over-compensated by the increased length of the other edges that were originally incident to it. Under this operation, the total number of blocks per design does not change. The total number of intersections is reduced by one after each edge-contraction. The mean distance between intersections tends to increase as more edges are contracted: it is not surprising because the total street length rises and the total number of edges drops (and there are no dead-end streets in our system).

The impact of the operation on the directional reach/distance: Initially, this operation tends to add directional distance between locations as more edges are contracted. However, after a certain point, the effect of this operation on the directional distance becomes much weaker and obscure, as evidenced by the distribution of the mean DDL20d per design, the mean linear reach per design, and the mean 2-dc reach per design.

The impact of the operation on the regularity of a street network: Contracting edges tends to create more irregular designs, as indicated by the measure of fragmentality. It tends to make less compact blocks, as indicated by the decreasing mean standardized area-perimeter ratio.

The impact of the operation on the diversity of syntactic conditions: Contracting edges tends to increase the diversity of syntactic conditions in a superblock design, as indicated by the increasing number of unique DDL20d values per design. The standard deviation of DDL20d values per design also tends to increase as more edges are contracted, although it tends to stabilize after a certain number of edges have already been contracted.

The effects and trends may not always hold as we apply this operation T times with T greater than the maximum number of times of application we have tested here. The effects discussed above are also dependent on the parameter settings used to generate the sample of designs (e.g., r_{min} and r_{max}) and to conduct the directional analysis (e.g., the threshold degree).

Part B: Cross-Concatenating Vertices

5.5 Operation: Cross-Concatenate Vertices

5.5.1 Definition of operation

The operation of cross-concatenating vertices joins two vertices *A* and *B* that are not adjacent to each other but belong to the same cell. In doing so, it merges the two vertices into a new vertex *C* that is adjacent to all vertices that were originally adjacent to either *A* or *B* (Figure 69).

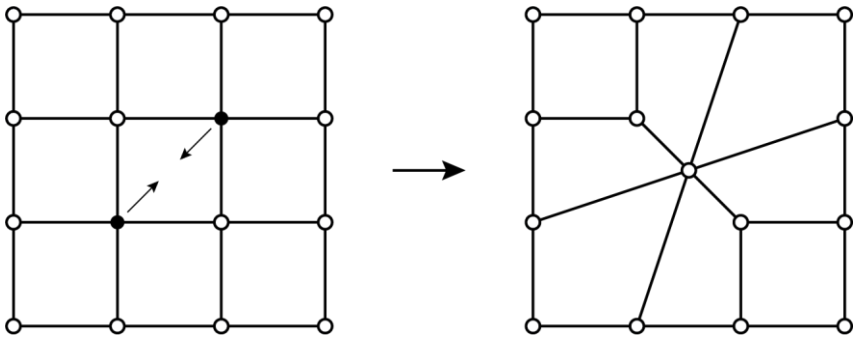

Figure 69. Cross-concatenate two vertices.

5.5.2 Parameters

To precisely control the position of the new vertex into which the two vertices are merged, we require that the new vertex *C* lies at a specified distance away from *A* towards the vertex *B*, as shown in Figure 70. Instead of specifying a fixed distance $d(A, C)$, we can specify the ratio between $d(A, C)$ and $d(A, B)$. Since the vertices *A* and *B* belong to the same cell, we can use the target vertices of two half edges h_1 and h_2 in a common cell to indicate the two vertices *A* and *B*, respectively.

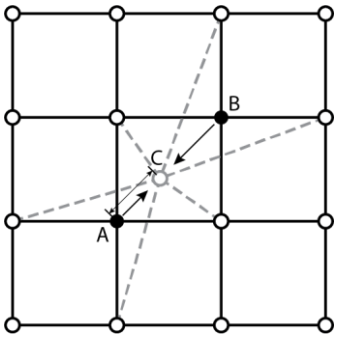

Figure 70. Parametric control of cross-concatenating vertices.

5.6 Generative Algorithm

5.6.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock designs, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by X;
- the number of streets running horizontally in the initial regular grid, denoted by Y;
- the first vertex to be concatenated, as indicated by a half-edge h_1;
- the second vertex to concatenated, as indicated by a half-edge h_2;
- the minimum ratio between the distance from the new vertex (resulting from the cross-concatenation) to $h_1.v$ and the distance from $h_1.v$ to $h_2.v$, denoted by r_{min};

- the maximum ratio between the distance from the new vertex (resulting from the cross-concatenation) to $h_1.v$ and the distance from $h_1.v$ to $h_2.v$, denoted by r_{max}; and
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled by the following parameters: l_x, l_y, X, Y, h_1, h_2, r_{min}, r_{max}, T.

5.6.2 General description

We start from a street graph that represents a regular grid. Then, we randomly choose a cell c inside the boundary of the street graph. We further randomly pick two half-edges h_1 and h_2 that belong to the same cell but not next to each other. If the chosen cell contains neither dangling edges nor bridges and the two vertices to be cross-concatenated (i.e., $h_1.v$ and $h_2.v$) do not lie on the border of the street graph, then we attempt the operation of cross-concatenating vertices. We continue to do so until the operation has been successfully performed a specified number of times.

To cross-concatenate two vertices A and B, we can first add an edge $\{A, B\}$ that connects A and B, and then contract the edge $\{A, B\}$. This way, we can build the operation of cross-concatenating vertices based on the operation of contracting edge which we have already defined before.

Four different scenarios are considered, as illustrated in Figure 71. In the first scenario, there are at least two intervening vertices between the first vertex to be concatenated, or $h_1.v$, and the second vertex to be concatenated, or $h_2.v$. This is true both

when we start from $h_1.v$ and count towards $h_2.v$ counterclockwise and when we start from $h_2.v$ and count towards $h_1.v$ counterclockwise. In the second scenario, there are at least two intervening vertices between the two vertices to be concatenated when we start from $h_2.v$ and count towards $h_1.v$ counterclockwise but only one intervening vertex when we start from $h_1.v$ and count towards $h_2.v$ counterclockwise. In the third scenario, there are at least two intervening vertices between the two vertices to be concatenated when we start from $h_1.v$ and count towards $h_2.v$ counterclockwise but only one intervening vertex when we start from $h_2.v$ and count towards $h_1.v$ counterclockwise. In the fourth scenario, there is exactly one intervening vertex between $h_1.v$ and $h_2.v$, either when we start from $h_1.v$ and count towards $h_2.v$ or start from $h_2.v$ and count towards $h_1.v$.

Note that although Scenarios 1, 2, and 3 would never occur by only applying the operation of cross-concatenating vertices on the initial 9 × 9 square-grid design, we still consider those scenarios so that the proposed algorithm can be used in more general situations.

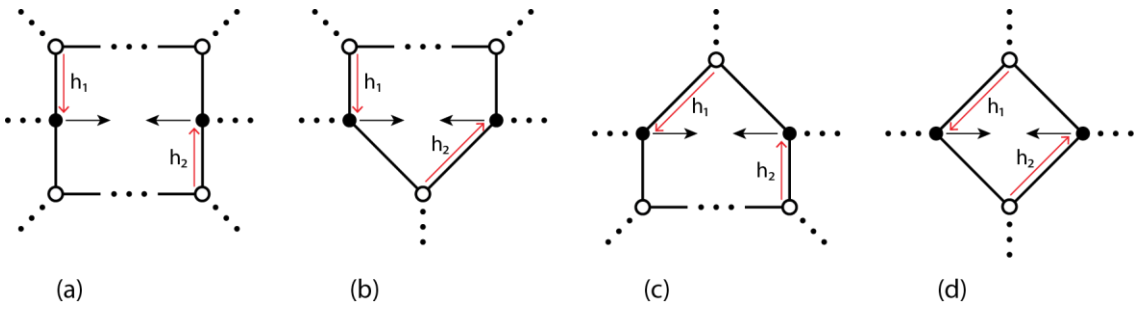

Figure 71. Different scenarios of cross-concatenating vertices.

5.6.3 Pseudocode

Suppose that we have already generated the initial street graph G which represents a regular grid that is l_x units long, l_y units wide, with X number of streets running vertically and Y number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of cross-concatenating vertices based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random* (a, b) refers to a procedure which returns a random floating-point number N such that $a \leqslant N \leqslant b$ for $a \leqslant b$, and *random_choice* (s) refers to a procedure which returns a random element from the set s.

Note that all the restrictions we set for the operations of adding edge (or linking vertex to vertex which will be introduced in Chapter 6) and contracting edge also apply here. If any of the operations are not successfully performed, it will restore the street graph to the state at the end of the previous attempt and start a new attempt. Figure 72A illustrates a successful application of the operation of cross-concatenating vertices, while Figure 72B illustrates a failed attempt to cross-concatenate the two vertices because it violates the specified conditions for edge-contraction.

ALGORITHM 5.2: Cross-concatenate Vertices

```
procedure cross_concatenate_vertices (G, r_min, r_max, T)
counter ≔ 0
while counter < T
        c ≔ random_choice (C)
        if c does not have any dangling edges or bridges inside then
                H_c ≔ a set comprising all half-edges that belong to c
                h_1 ≔ random_choice (H_c)
                if h_1.v does not lie on the border of G then
                        H_c' ≔ H_c – {h_1, h_1.prev_h, h_1.next_h}
                        h_2 ≔ random_choice (H_c')
                        if h_2.v does not lie on the border of G then
                                r ≔ random (r_min, r_max)
                                p ≔ (h_1.v.x + r (h_2.v.x – h_1.v.x), h_1.v.y + r (h_2.v.y – h_1.v.y))
                                /* scenario 1 */
                                if h_1.next_h.next_h is not h_2 and h_2.next_h.next_h is not h_1
                                then
                                        add the edge {h_1.v, h_2.v}
                                        contract the edge {h_1.v, h_2.v} to the point p
                                        counter ≔ counter + 1
                                /* scenario 2 */
                                if h_1.next_h.next_h is h_2 and h_2.next_h.next_h is not h_1 then
                                        remove the edge h_1.next_h.e
                                        add the edge {h_1.v, h_2.v}
                                        contract the edge {h_1.v, h_2.v} to the point p
                                        counter ≔ counter + 1
                                /* scenario 3 */
                                if h_1.next_h.next_h is not h_2 and h_2.next_h.next_h is h_1 then
                                        remove the edge h_1.e
                                        add the edge {h_1.twin_h.prev_h.v, h_2.v}
                                        contract the edge {h_1.twin_h.prev_h.v, h_2.v} to the
                                        point p
                                        counter ≔ counter + 1
                                /* scenario 4 */
                                if h_1.next_h.next_h is h_2 and h_2.next_h.next_h is h_1 then
                                        remove the edge h_1.next_h.e
                                        add the edge {h_1.v, h_2.v}
                                        remove the edge h_1.e
                                        contract the edge {h_1.v, h_2.v} to the point p
                                        counter ≔ counter + 1
```

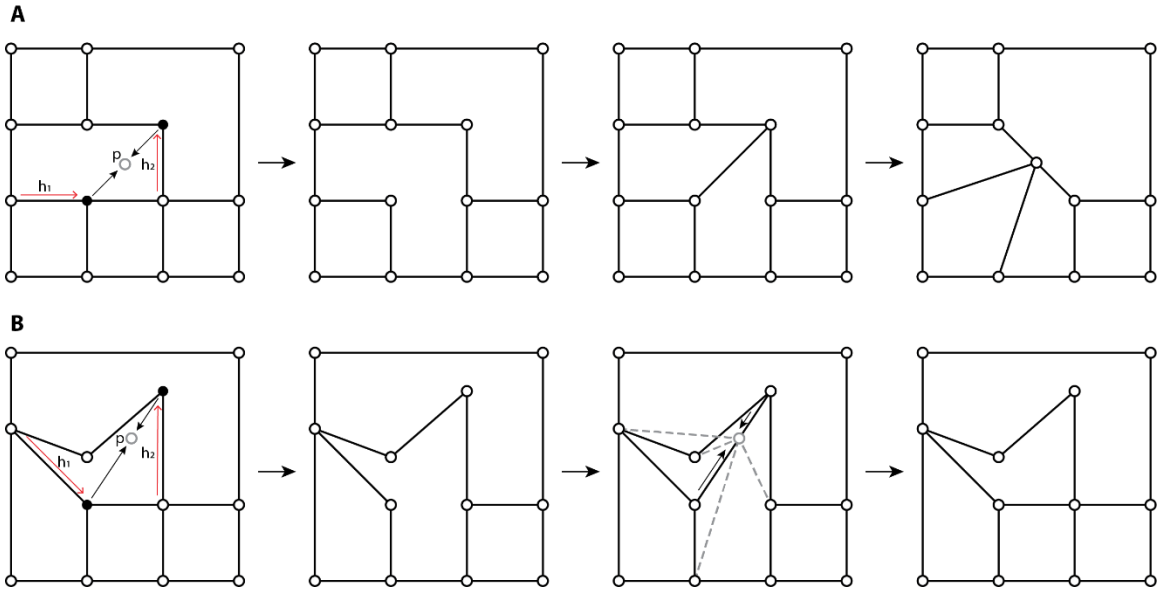

Figure 72. Examples of successful and unsuccessful application of the operation of cross-concatenating vertices (cross-concatenate vertices). A: Successful application of operation. B: Unsuccessful application of operation.

5.7 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the six groups of designs, we varied only one control parameter—namely T, the total number of operations to be performed—but kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9, r_{min} = 0.5, r_{max} = 0.5. By assigning the same value (0.5) to both r_{min} and r_{max}, we require that the vertices, after concatenation, become a single vertex located midway between the two vertices being concatenated.

5.7.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation (i.e., the appropriate value of T) on the

initial 9 × 9 square-grid design. We adopted the following strategy: First, we choose an arbitrarily high number a so that it is impossible to apply the operation of cross-concatenating vertices more than a times on the initial square-grid design based on our algorithm. We run the algorithm by setting $T = a$ and a terminating condition: If it has made 2000 attempts in a row and none of them are successful, then it will stop attempting the operation anymore and record the total number of times that the operation has been successfully applied, denoted by b. This gives us a sense of the upper limit for the number of successful applications of the operation for a particular run. However, the recorded number b may vary for different runs. Thus, we make 100 trial runs and record the number b for each trial. Then we choose the minimum recorded value for b among all the trial runs as the maximum number of times we would apply this operation to generate a group of designs, denoted by T_{max}. The minimum number of times that we would apply the operation, T_{min}, is subsequently determined by finding the nearest integer for $T_{max} / 6$. To generate the other four groups of designs, we simply set T to $2T_{min}$, $3T_{min}$, $4T_{min}$, $5T_{min}$, respectively.

Examples of designs from each group are shown in Figure 73. Note that the blocks (cells) consist of only quads and triangles.

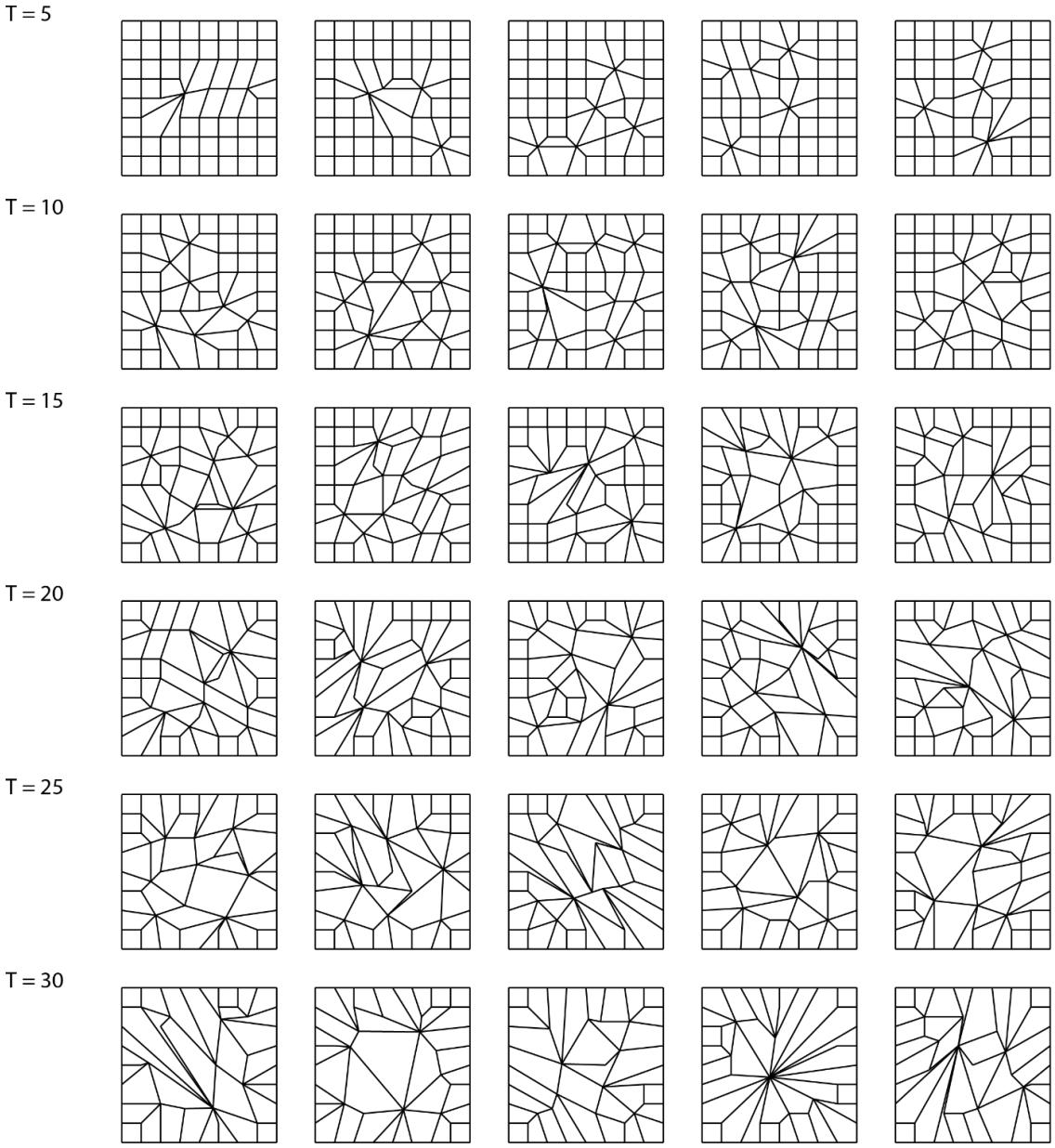

Figure 73. Examples of designs from each group (cross-concatenate vertices).

5.7.2 Data analysis

Six groups of designs are generated by applying the operation of cross-concatenating vertices 5, 10, 15, 20, 25, and 30 times, respectively. Each group consists of 100 designs. The different groups of designs are analyzed and compared based on

measures that characterize distinct aspects of designs. We analyzed the measures related to the graph properties based on the intermediate street graphs. We analyzed all the other measures based on the final street graphs (by "polishing" the intermediate street graphs).

Elementary graph properties

(1) Number of vertices

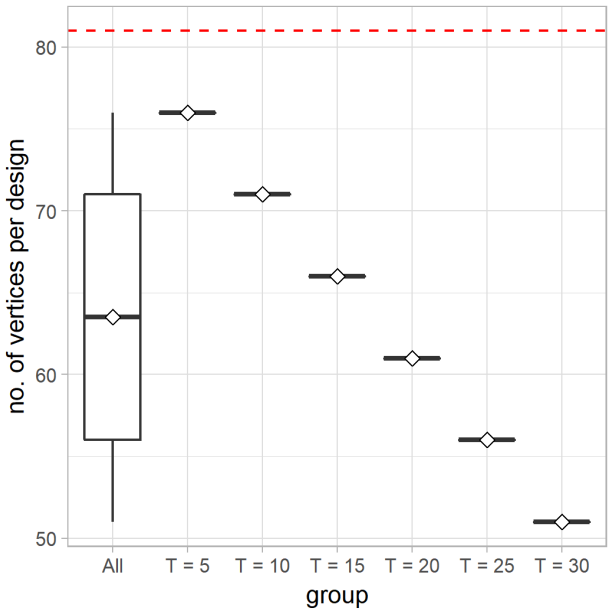

Figure 74. A boxplot (with mean diamonds) showing the distribution of the total number of vertices per design for all designs and for each group of designs (cross-concatenate vertices). The red dashed line indicates the total number of vertices in the initial 9 × 9 square-grid design (81).

By definition, the operation of cross-concatenation merges two vertices into one, thus reducing the total number of vertices by one. As shown in Figure 74, the total number of vertices per design drops consistently as more vertices are cross-concatenated during the generative process. For a design generated by applying the operation of cross-

concatenating vertices T times on the initial street graph, the total number of vertices, or $|V|$, equals $81 - T$.

(2) Number of edges

As shown in Figure 75, the total number of edges per design drops steadily as the operation of cross-concatenating vertices is more frequently applied. Based on our algorithm and the initial setup of the street graph (the 9 × 9 square-grid), the operation is always applied in the context of Scenario 4 (see sections 5.2.2 and 5.2.3). As a result, the total number of edges is reduced by two every time the operation is successfully applied. For a design generated by applying the operation of cross-concatenating vertices T times on the initial street graph, the total number of edges, or $|E|$, equals $144 - 2T$.

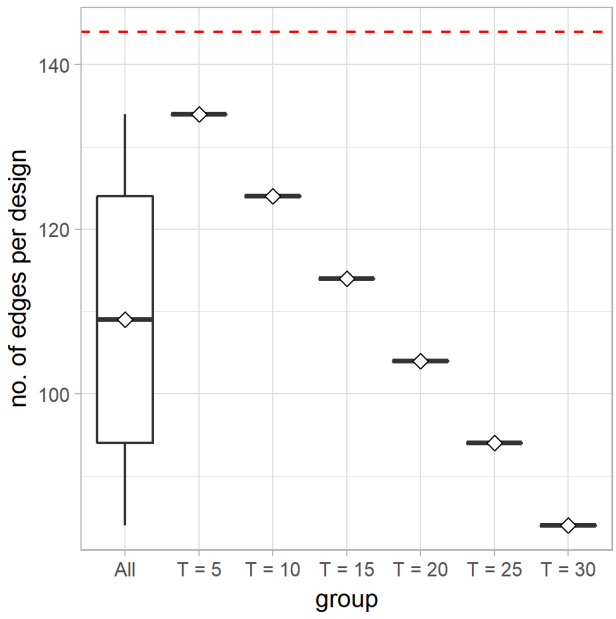

Figure 75. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (cross-concatenate vertices). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

As shown in Figure 76, the total number of cells per design drops steadily as the operation of cross-concatenating vertices is applied more frequently. Based on the generative algorithm we designed and the initial setup of the street graph (i.e., a 9 × 9 square-grid), the total number of cells is reduced by one every time the operation is successfully applied. For a design generated by applying the operation of cross-concatenating vertices T times on the initial street graph, the total number of cells, or $|C|$, equals $64 - T$.

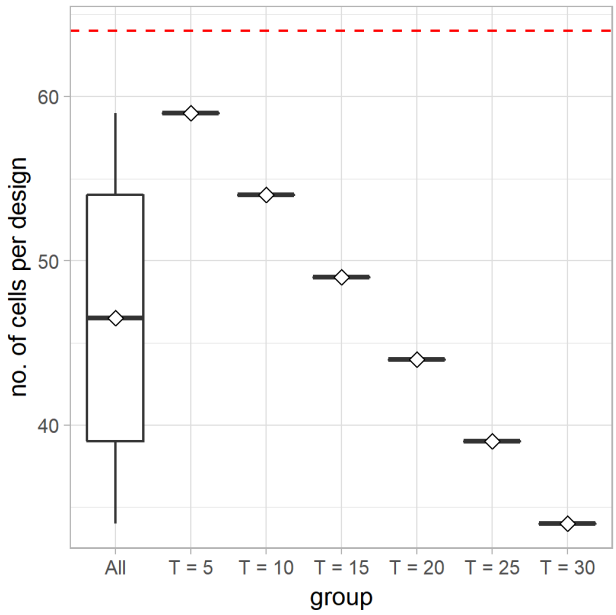

Figure 76. A boxplot (with mean diamonds) showing the distribution of the total number of cells per design for all designs and for each group of designs (cross-concatenate vertices). The red dashed line indicates the total number of cells in the initial 9 × 9 square-grid design (64).

(4) Vertex degree

As shown in Figure 77A, the vertex degree ranges from 2 to 14. The distribution of the vertex degree is heavily right-skewed. The most common vertex degree for all the vertices analyzed is 3, and the second most common vertex degree is 4. Extremely high vertex degrees tend to occur in the groups of designs generated by more frequent application of the operation of cross-concatenating vertices (Figure 77B).

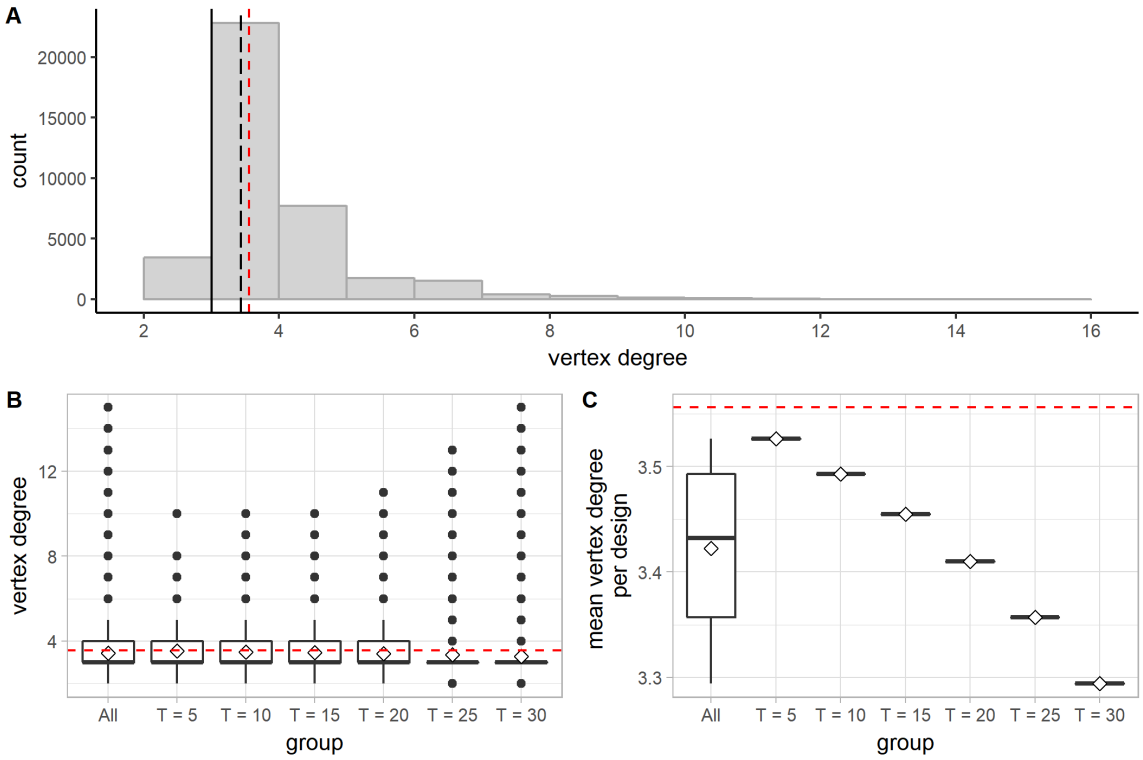

Figure 77. Distribution of the vertex degree (cross-concatenate vertices). A: A histogram showing the distribution of the vertex degree for all vertices in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design (≈ 3.556). B: A boxplot (with mean diamonds) showing the distribution of the vertex degree for all vertices and for vertices in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs.

However, the mean vertex degree per design drops steadily for the groups of designs generated by more frequent application of this operation (Figure 77C). Based on the handshaking theorem, the mean vertex degree for the designs generated by applying the operation T times on the initial 9 × 9 square-grid design equals $2(144 - 2T) / (81 - T)$.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

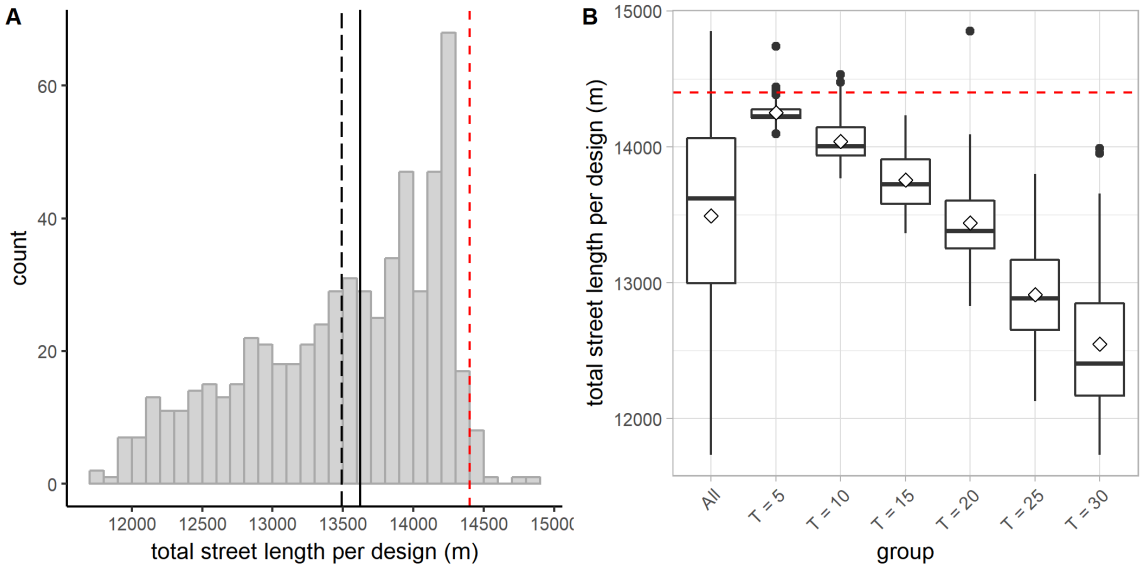

Figure 78. Distribution of the total street length per design (cross-concatenate vertices). A: A histogram showing the distribution of the total street length per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs.

As shown in Figure 78, the total street length per design ranges from slightly under 12000 m to nearly 15000 m. The vast majority of designs have a total street length of less than 14400 m—that is, less than the total length of streets in the initial 9 × 9

square-grid design. As shown in Figure 78B, the total street length per design tends to decrease for the designs generated by more frequent cross-concatenation of vertices. As *T* increases, the total street length per design also tends to cover an increasingly wider interquartile range of values.

(2) Total number of blocks per design

The total number of blocks per design is equal to the total number of cells per design. Therefore, just like the total number of cells per design, the total number of blocks per design decreases as the operation of cross-concatenating vertices is applied more frequently.

(3) Total number of intersections per design

As shown in Figure 79, the total number of intersections per design tends to drop for the designs generated by more frequent cross-concatenation of vertices. For a design generated by applying the operation *T* times on the initial street graph, the total number of intersections would be reduced by at least *T*. The total number of intersections would be reduced by more than *T* in cases where the operation produces vertices that have degree two which, by definition, are not counted as intersections.

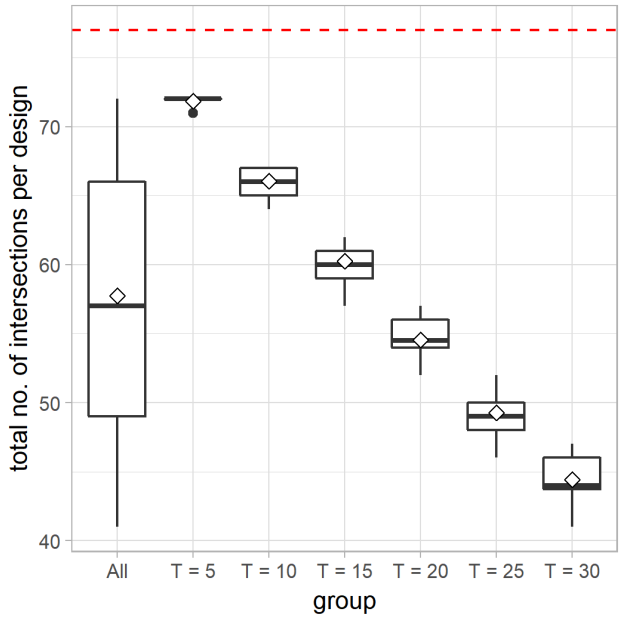

Figure 79. A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs (cross-concatenate vertices). The red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77).

(4) Distance between intersections

As shown in Figure 80, the distance between intersections in the designs generated can be shorter than 30 m or longer than 650 m. Not surprisingly, in most cases the distance between two intersections is 100 m. Longer distances between intersections occur more often in the groups of designs generated by more frequent cross-concatenation of vertices (Figure 80B). As shown in Figure 80C, as *T* increases, the mean distance between intersections per design tends to increase, and it also tends to cover an increasingly wider interquartile range of values.

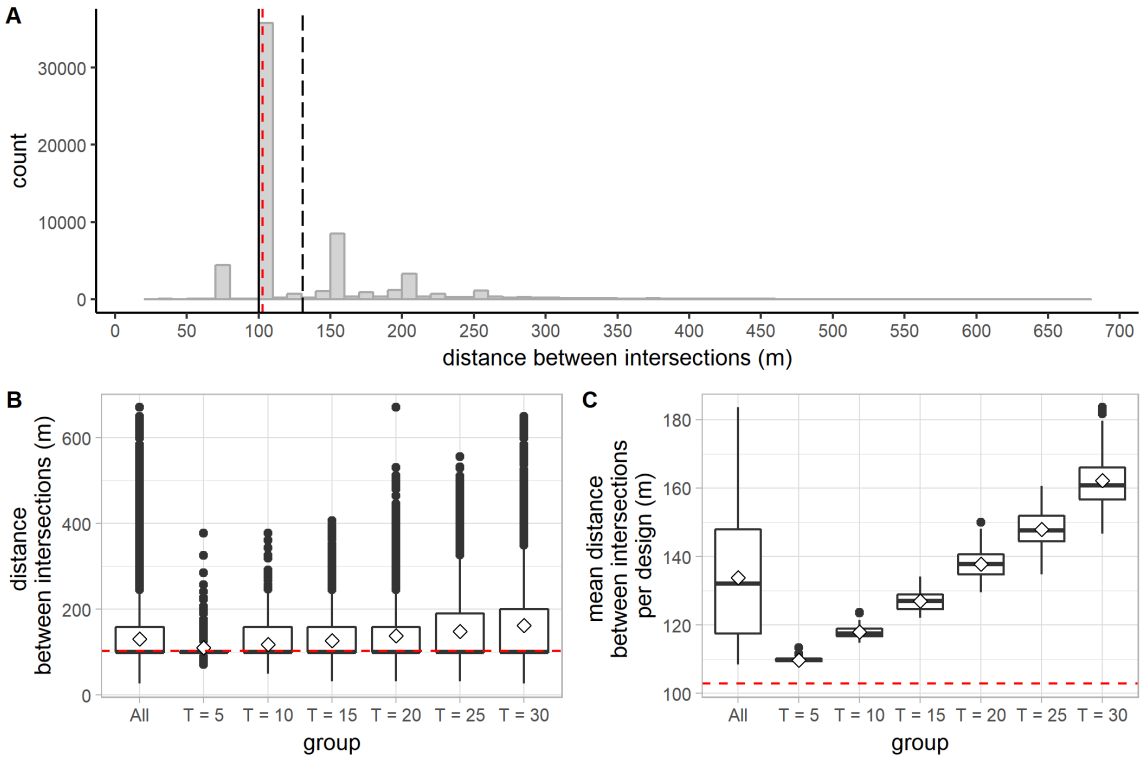

Figure 80. Distribution of the distance between intersections (cross-concatenate vertices). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the distance between intersections observed in all designs and in each group of designs. C: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Directional reach and directional distance

(1) DDL

As shown in Figure 81, the distribution of the DDL values varies when the threshold angle is set differently for the directional reach analysis. The mean DDL for all segments in all designs drops significantly as the threshold angle is increased from 10° to

30°. In the following analysis, unless otherwise specified, the directional distance is always evaluated by setting the threshold angle to 20°.

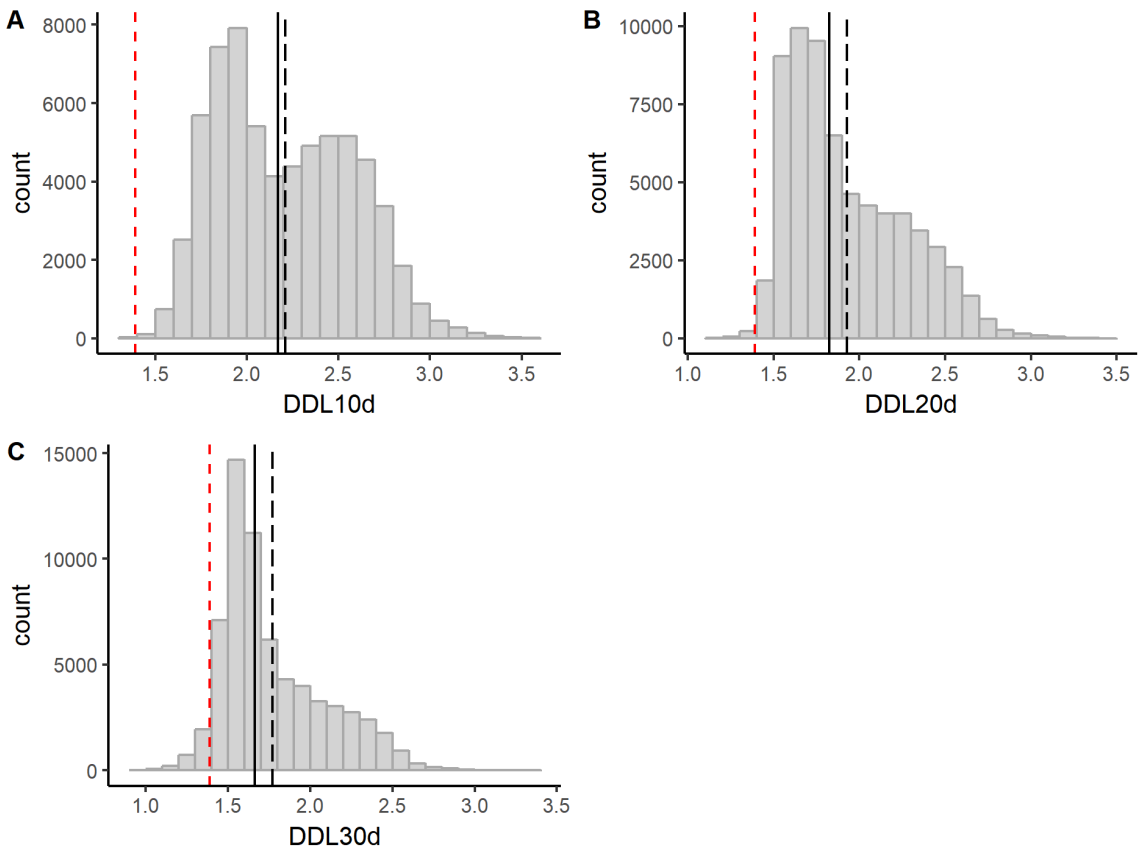

Figure 81. Distributions of DDL10d, DDL20d, and DDL30d (cross-concatenate vertices). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

As shown in Figure 81B, the distribution for DDL20d is skewed to the right. The value of DDL20d for all the segments analyzed ranges from 1.1 to 3.5. More than half of the segments assume a DDL20d value that is lower than 1.9. As shown in Figure 82A, the segments in the groups of designs generated by more frequent application of operation generally assume a wider range of values. More importantly, as the operation is

applied more frequently, both very high and very low DDL20d values are observed—in other words, the application of the operation of cross-concatenating vertices does not just push the extreme DDL20d values in a single direction but produces both unusually integrated and unusually segregated segments. As *T* increases from 5 to 15, the mean DDL20d per design tends to increase. However, a downward trend starts to appear as *T* increases after that point.

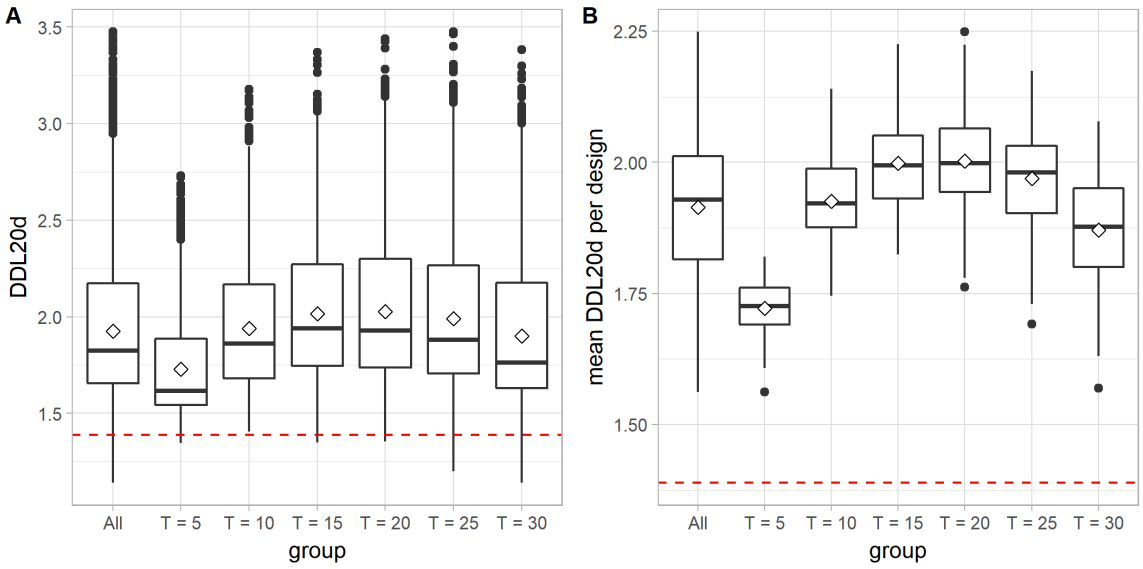

Figure 82. Distribution of DDL20d (cross-concatenate vertices). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 83, the linear reach for all the segments analyzed ranges from below 50 m to more than 3500 m. About half of all segments have a linear reach of more than or equal to 600 m. More than 40% of all segments have a linear reach of between

800 and 850 m—that is, about the same length as a full side of the superblock. When analyzed in groups, as *T* increases from 5 to 15, the mean linear reach per design tends to decrease. However, an upward trend starts to appear as *T* increases after that point.

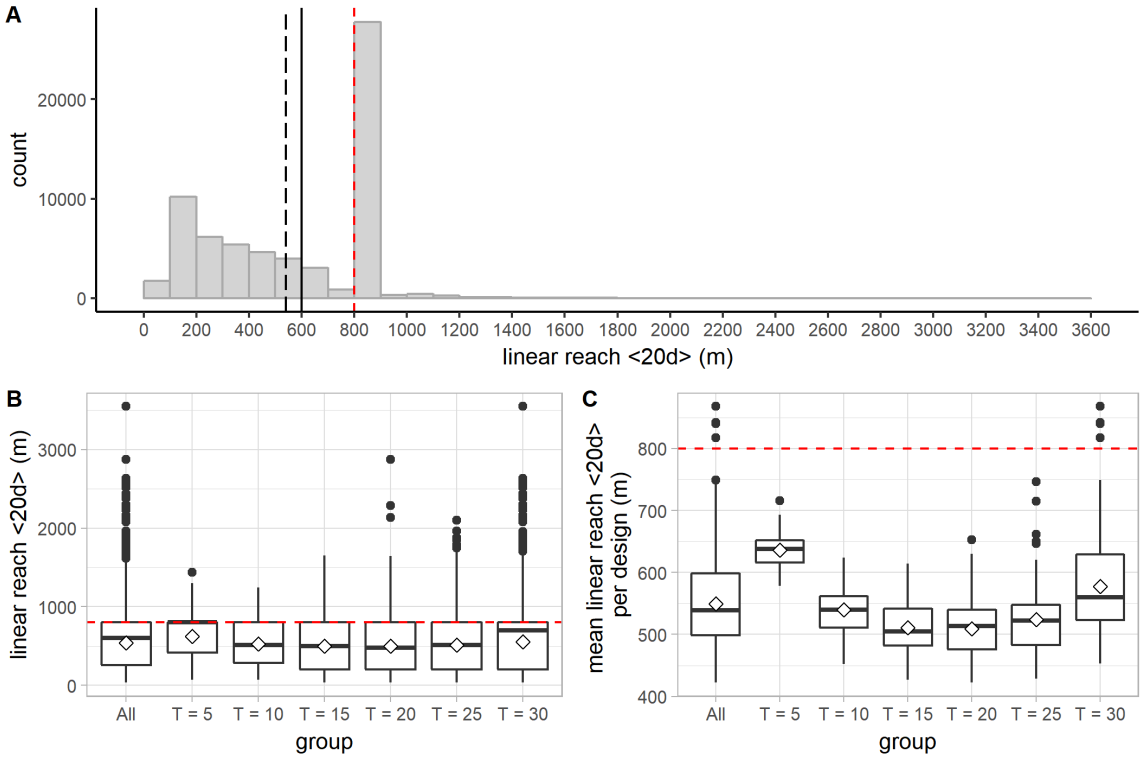

Figure 83. Distribution of the linear reach (cross-concatenate vertices). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

(3) 2-dc reach (dr2dc20d)

As shown in Figure 84A, the distribution for 2-dc reach is skewed to the left. The value of 2-dc reach for all the segments analyzed ranges from around 2000 m to nearly

15000 m. More than half of all the segments have a 2-dc reach of greater than 11000 m. When analyzed in groups, as *T* increases from 5 to 20, the mean 2-dc reach per design tends to decrease. However, the downward trend disappears as *T* increases after that point (Figure 84C).

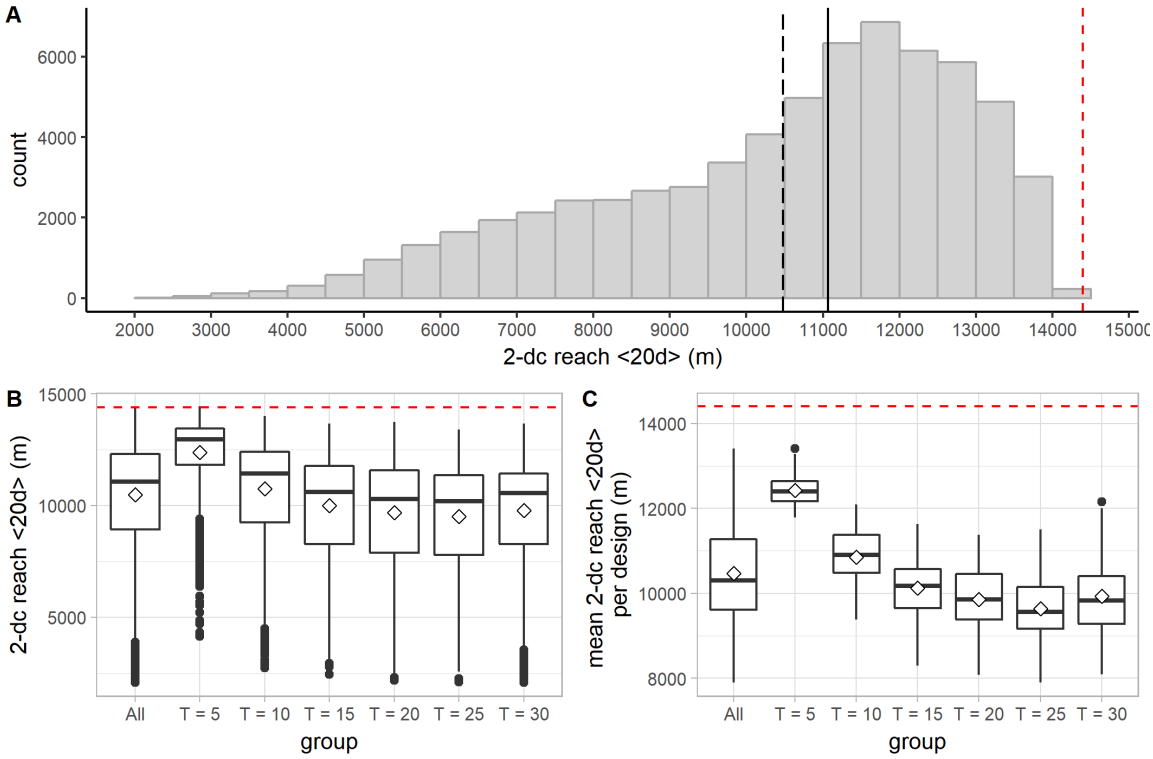

Figure 84. Distribution of the 2-dc reach (cross-concatenate vertices). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

As shown in Figure 85, the fragmentality per design ranges from 0.19 to 0.56. Nearly 60% of all designs have a fragmentality value greater than 0.40. When analyzed in groups, as *T* increases, the fragmentality per design also tends to increase.

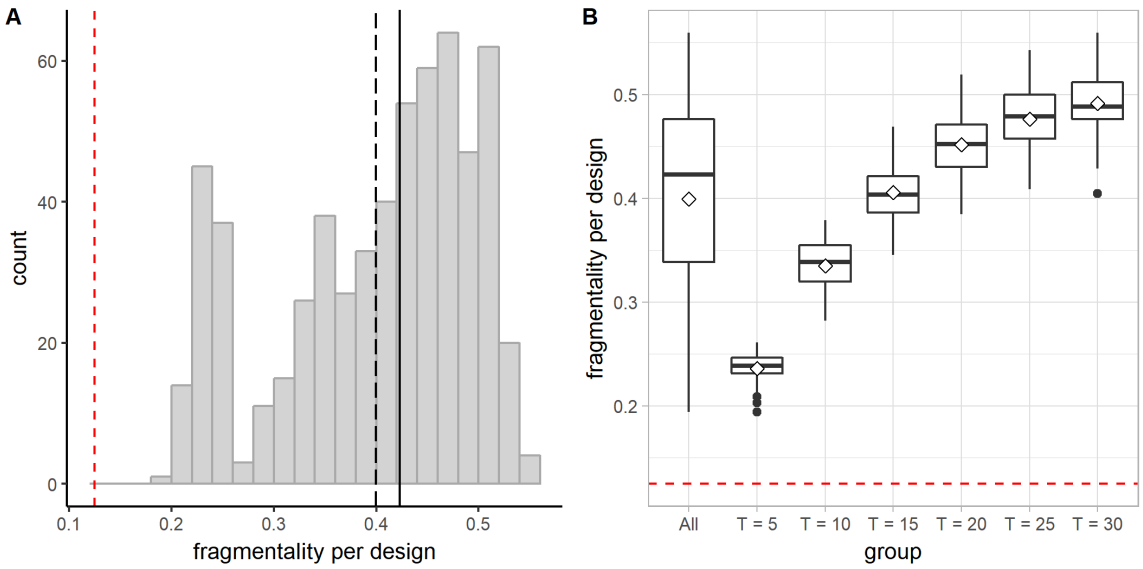

Figure 85. Distribution of the fragmentality per design (cross-concatenate vertices). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

As shown in Figure 86, the distribution for the block area is heavily right-skewed. The area of a block ranges from slightly under 2000 m² to more than 100000 m². In other words, it could be smaller than a fifth of a 100 × 100 m block, or ten times as large as a

100 × 100 m block. About 55% of all blocks are of exactly the same size as the 100 × 100 m block. The blocks in the groups of designs generated by more frequent application of operation assume a wider range of values (Figure 86B). When analyzed in groups, as *T* increases, the mean block area per design increases steadily. Since the number of blocks for a design generated by applying the operation *T* times on the initial 9 × 9 square-grid design is 64 – *T*, it follows that its mean block area equals 800 × 800 / (64 – *T*) m², or 640000 / (64 – *T*) m².

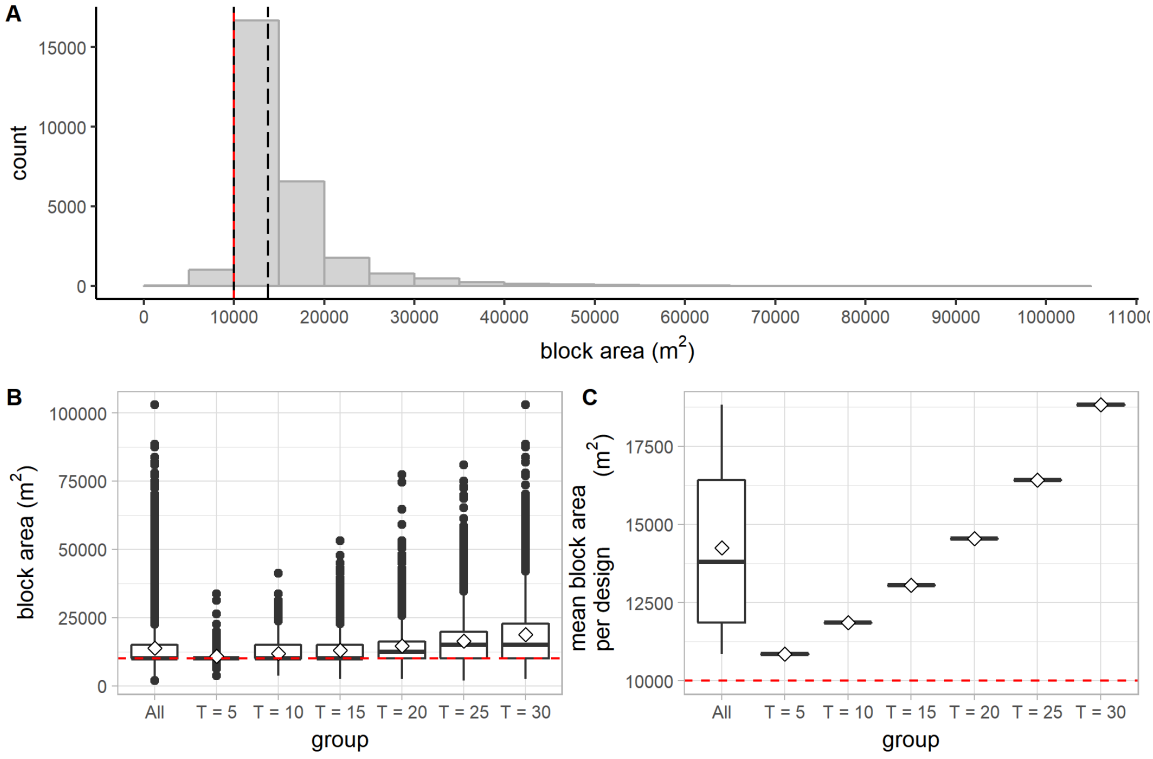

Figure 86. Distribution of the block area (cross-concatenate vertices). A: A histogram showing the distribution of the area for all the blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²). B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs. C: A boxplot showing the distribution of the mean block area per design for all designs and for each group of designs.

The areas of blocks inside a design become more and more varied as the operation is applied more and more frequently, as evidenced by the increasing standard deviation of the block area per design (Figure 87).

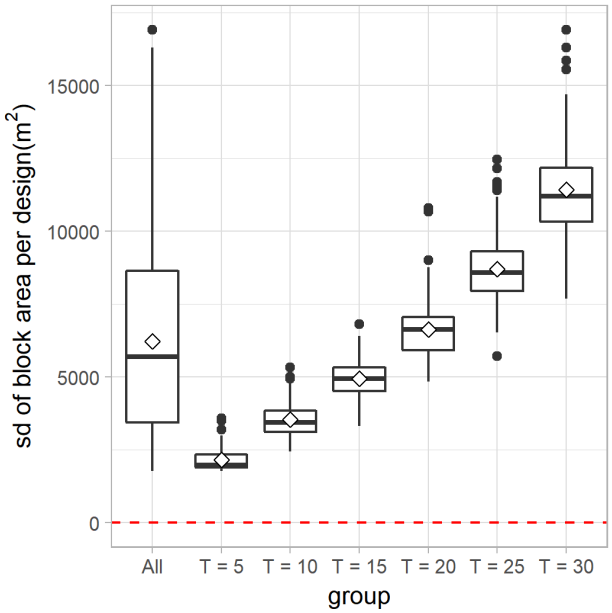

Figure 87. A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs (cross-concatenate vertices). The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²).

(3) Block perimeter

As shown in Figure 88, the distribution for the block perimeter is heavily right-skewed. The perimeter of a block ranges from around 350 m to 1550 m. Nearly 60% of all blocks analyzed have a perimeter length of smaller than 500 m. The blocks with extremely large perimeter tend to occur in the designs generated by more frequent application of operation. The values for the block perimeter are also more varied for the groups of designs generated by more frequent cross-concatenation of vertices, as

evidenced by the increasingly wider interquartile ranges (Figure 88B). When analyzed in groups, as *T* increases, the mean block perimeter per design tends to increase as well (Figure 88C).

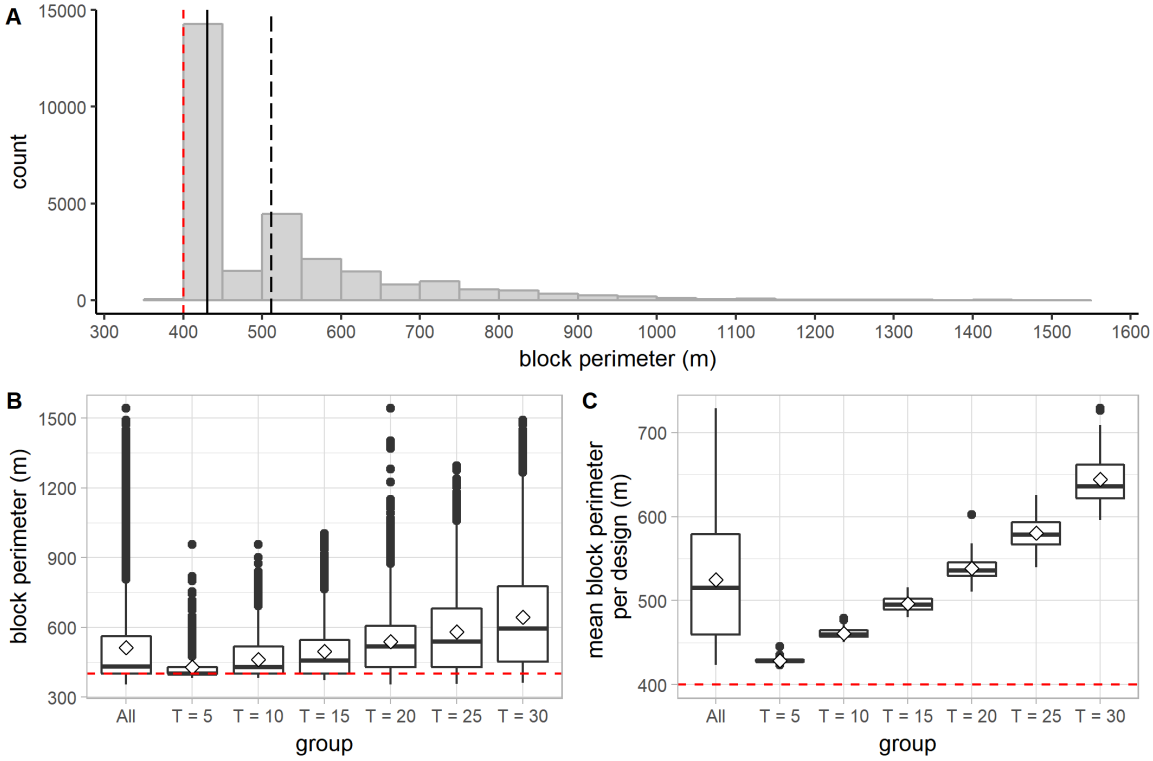

Figure 88. Distribution of the block perimeter (cross-concatenate vertices). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

The perimeters of blocks inside a design become more and more varied as the operation is applied more and more frequently, as evidenced by the increasing standard deviation of the block perimeter per design (Figure 89).

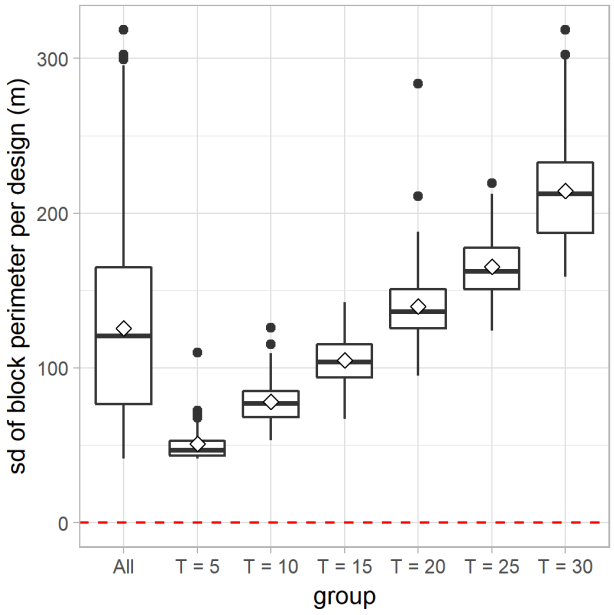

Figure 89. A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs (cross-concatenate vertices). The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m).

(4) Standardized block area-perimeter ratio (SAPR)

As shown in Figure 90, the standardized block area-perimeter ratio (SAPR) ranges from 0.25 to 1.00. More than 70% of all blocks analyzed have a SAPR greater than 0.90. The blocks with extremely low SAPR tend to occur in the designs generated by more frequent application of this operation (Figure 90B). When analyzed in groups, as *T* increases, the mean block SAPR per design tends to decrease as well (Figure 90C).

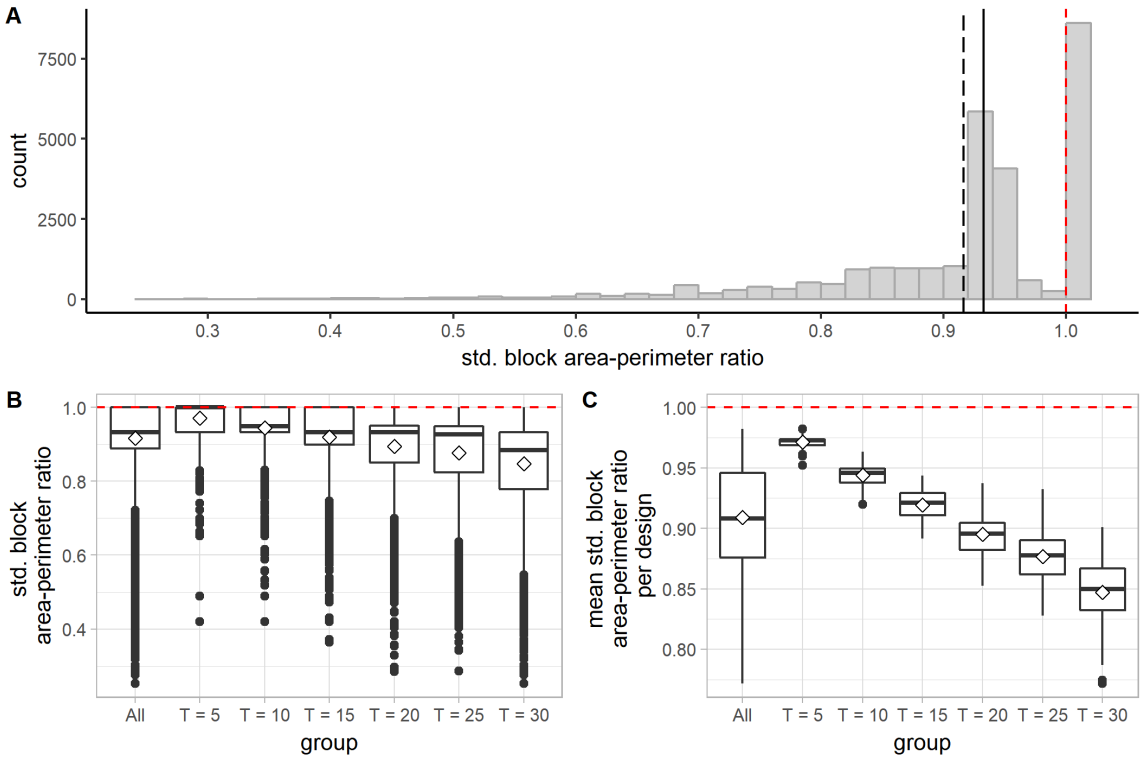

Figure 90. Distribution of the standardized block area-perimeter ratio (SAPR) (cross-concatenate vertices). A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and proportion of distinct DDL20d values per design

As shown in Figure 91, the total number of distinct DDL20d values per design ranges from 23 to 57. More than half of all designs have 40 or more distinct DDL20d values per design. More frequent application of the operation of cross-concatenating vertices tends to increase the total number of distinct DDL20d values (or syntactic

conditions) as *T* increases from 5 to 20 (Figure 91B). The trend becomes more apparent when we take the proportion of distinct values instead of the absolute number so that we can counteract the effect caused by the reduction of the total number of segments available. As shown in Figure 91C, the proportion of distinct DDL20d values per design ranges from 0.17 to 0.60, and it tends to increase as *T* increases.

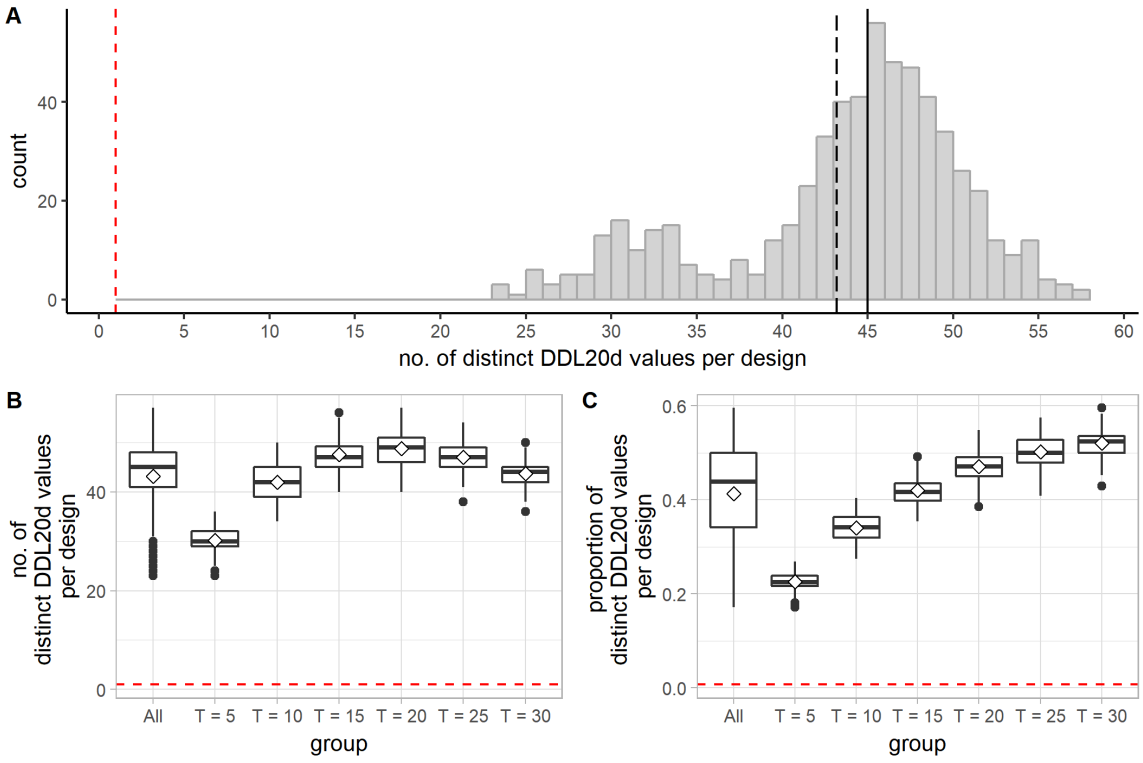

Figure 91. Distribution of the total number and the proportion of distinct DDL20d values per design (cross-concatenate vertices). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d per design

As shown in Figure 92, the standard deviation for DDL20d per design ranges from 0.19 to 0.42. When analyzed in groups, as T increases, the standard deviation for DDL20d per design tends to increase (although the upward trend becomes less pronounced after a certain point).

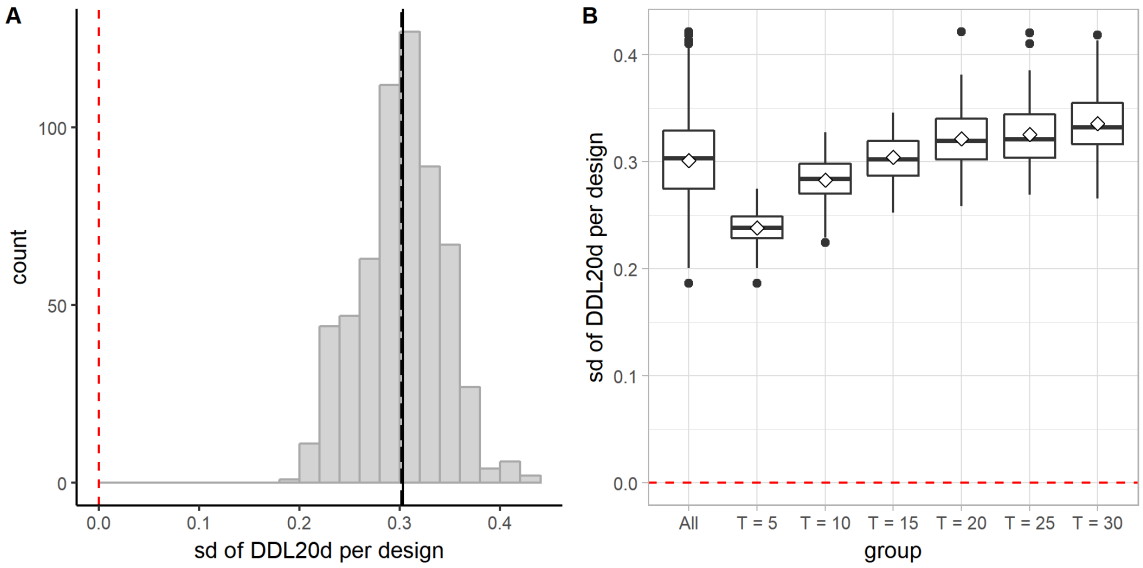

Figure 92. Distribution of the standard deviation of DDL20d per design (cross-concatenate vertices). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

5.8 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: Based on our algorithm, a single application of the operation of cross-concatenating vertices would reduce the total number of vertices by one, reduce the total number of edges by two, and reduce the total number of cells by one. Although this operation can produce vertices of extremely high degrees, the mean vertex degree per design decreases as the operation is applied more frequently, showing an opposite trend compared to the operation of contracting edge.

The impact of the operation on the density of a street network: This operation tends to reduce the total street length per design, which is, again, contrary to the trend observed for the operation of contracting edge. The result is a balance between the loss of edges and the increased length of the incident edges because of the operation. The total number of blocks declines because of the operation, and the total number of intersections per design also tends to decrease as more vertices are cross-concatenated. The mean distance between intersections per design tends to increase. So far, we have seen at least two different interplays between total street length per design and the mean distance between intersections per design: (1) total street length increases and mean distance between intersections increases, as in the case of shifting vertex and contracting edge; (2) total street length decreases and mean distance between intersections increases, as shown here.

The impact of the operation on the directional reach/distance: It tends to increase the mean directional distance between locations at the early stages. However, after being applied a certain number of times, the increasing trend disappears and, sometimes, an opposite trend begins to occur, as evidenced by the distributions of DDL20d, dr0dc20d, and dr2dc20d.

The impact of the operation on the regularity of a street network: The fragmentality per design tends to increase. The mean block area per design will be greater (because of the reduction of the total number of blocks per design), and the blocks sizes become more varied in a design. The mean block perimeter per design tends to be greater, and the perimeters of blocks in a design tend to be more varied. The shape of the average block tends to be less compact.

The impact of the operation on the diversity of syntactic conditions: It tends to increase the diversity of syntactic conditions in a superblock design, as evidenced by the overall increasing proportion of distinct DDL20d values per design as the operation is applied more frequently. The DDL20d values also tend to be more varied and dispersed for a design generated by more frequent applications of the operation of cross-concatenating vertices.

CHAPTER 6
LINKING VERTEX TO VERTEX: CREATING DIAGONAL STREETS

6.1 Operation: Link Vertex to Vertex

6.1.1 Definition of operation

The operation of linking vertex to vertex adds an edge to connect two existing vertices that belong to the same cell (Figure 93).

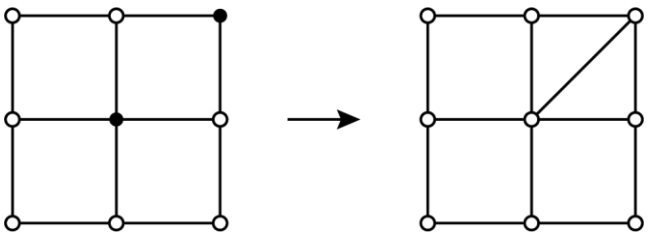

Figure 93. Link vertex to vertex.

6.1.2 Parameters

Once the vertices to be linked are specified, there is no ambiguity about how to perform the operation. Therefore, no parameters are needed.

6.2 Generative Algorithm

6.2.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock design, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by X;
- the number of streets running horizontally in the initial regular grid, denoted by Y;
- the first vertex to be linked, as indicated by a half-edge h_1;
- the second vertex to be linked, as indicated by a half-edge h_2;
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled by the following parameters: l_x, l_y, X, Y, h_1, h_2, T.

6.2.2 General description

We start from a street graph that represents a regular grid. Then we randomly pick a cell c in the street graph. We further randomly pick two half-edges, h_1 and h_2, which belong to the cell c. Finally, we attempt to link the vertex $h_1.v$ to the vertex $h_2.v$ with the edge $\{h_1.v, h_2.v\}$. We continue to do so until the operation has been successfully performed a specified number of times. The operation can only be performed if the following two conditions are met.

1. The edge to be added, $\{h_1.v, h_2.v\}$, does not intersect any existing edges at points other than $h_1.v$ and $h_2.v$.
2. The edge to be added is internal to the cell c (to which the two half-edges belong).

The conditions above are included to simplify the task of maintaining the integrity of the underlying data structure. Below illustrate a few situations where one of the above

conditions is not met. Note that the situations illustrated here are hypothetical and may be unlikely or impossible to occur by applying the operation alone on a regular grid. Nevertheless, we still check those conditions so that the proposed algorithm can be used in more general situations.

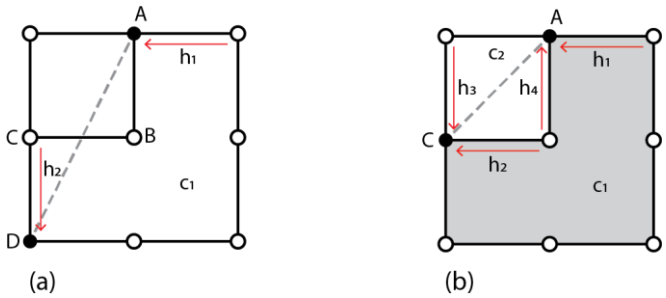

Figure 94. Examples of illegitimate moves (link vertex to vertex).

In the example shown in Figure 94a, suppose that the half-edges h_1 and h_2 in the cell c_1 are selected to attempt the operation of linking the vertex A ($h_1.v$) to the vertex D ($h_2.v$). The operation, however, is not allowed because the edge to be added, $\{A, D\}$, would intersect the existing edge $\{B, C\}$ at a point other than A and D—that is, a violation of Condition 1.

In the example shown in Figure 94b, suppose that the half-edges h_1 and h_2 in the cell c_1 are selected to attempt the operation of linking the vertex A ($h_1.v$) to the vertex C ($h_2.v$). The operation, however, is not allowed because the edge to be added, $\{A, C\}$, is external to the cell c_1 (lightly shaded in Figure 94b) in the sense that it lies outside the boundary of c_1 which h_1 and h_2 belong to. On the other hand, the operation would be allowed if the half-edges h_3 and h_4 were selected to link the vertex C to the vertex A

because, in that case, the edge $\{A, C\}$ would be internal to the cell c_2 which h_3 and h_4 belong to and it would not intersect any existing edges improperly.

6.2.3 Pseudocode

Suppose that we have already generated the initial street graph G which represents a regular grid that is l_x units long, l_y units wide, with X number of streets running vertically and Y number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of contracting edge based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random_choice* (*s*) refers to a procedure which returns a random element from the set *s*.

ALGORITHM 6.1: Link Vertices to Vertices

```
procedure link_vertices_to_vertices (G, T)
counter ≔ 0
while counter < T
        c ≔ random_choice (C)
        H_c ≔ a set comprising all half-edges that belong to c
        h_1 ≔ random_choice (H_c)
        H_c' ≔ H_c – {h_1, h_1.prev_h, h_1.next_h}
        h_2 ≔ random_choice(H_c')
        if the edge {h_1.v, h_2.v} does not intersect any existing edges at points other h_1.v and
        h_2.v
        and {h_1.v, h_2.v} is internal to c then
                add the edge {h_1.v, h_2.v}
                counter ≔ counter + 1
```

6.3 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the different groups of designs, we varied only one

control parameter—namely, T, the total number of operations to be performed—and kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9.

6.3.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation on the initial 9 × 9 square-grid design—that is, the values of T. It is easy to see that, based on our algorithm, we can only apply this operation on each square block in the original square-grid design at most once. Therefore, the maximum number of times that we can perform the operation of linking vertex to vertex based on the initial 9 × 9 square-grid design is the same as the total number of blocks in the initial design, that is, 64. This became the largest T value we used to generate the superblock designs. The minimum number of times to apply the operation, T_{min}, was subsequently determined by finding the nearest integer for 64 / 6, which is 11. Then we simply set T to integer multiples of 11 to generate the other groups of designs.

Examples of designs from each group are shown in Figure 95. Note that the blocks/cells consist of only quads and triangles.

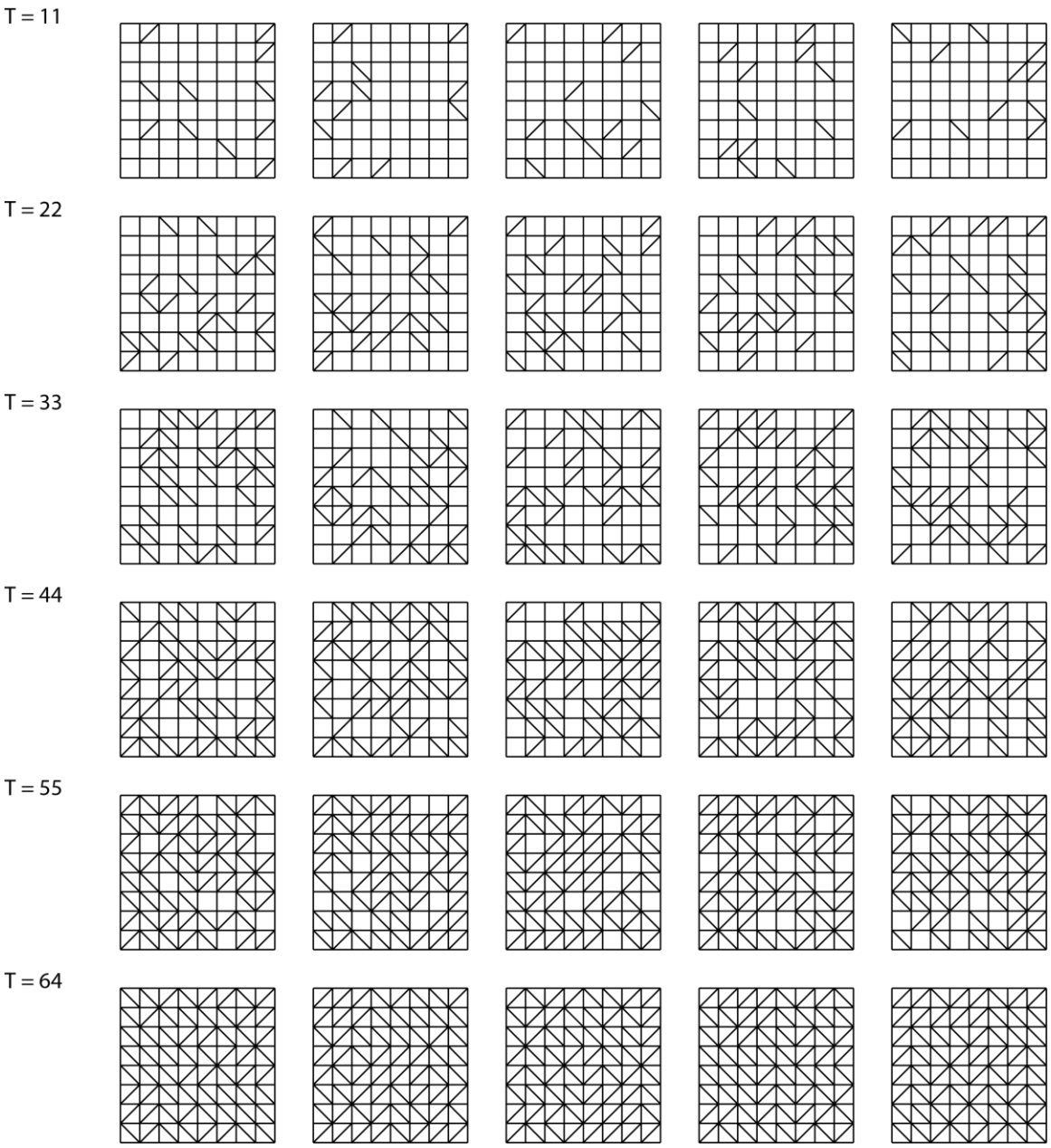

Figure 95. Examples of designs from each group (link vertex to vertex).

6.3.2 Data analysis

Six groups of designs were generated by applying the operation of linking vertex to vertex 11, 22, 33, 44, 55, and 64 times, respectively. Each group consists of 100 designs. The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of designs.

Elementary graph properties

(1) Number of vertices

By definition, the operation of linking vertex to vertex never creates new vertices. Therefore, the total number of vertices per design, or $|V|$, always equals 81.

(2) Number of edges

By definition, the operation of linking vertex to vertex adds an additional edge to the street graph. Therefore, as shown in Figure 96, the total number of edges per design increases as the operation is applied more frequently. More specifically, for a design generated by applying the operation T times on the initial 9×9 square-grid, we know that $|E| = 144 + T$.

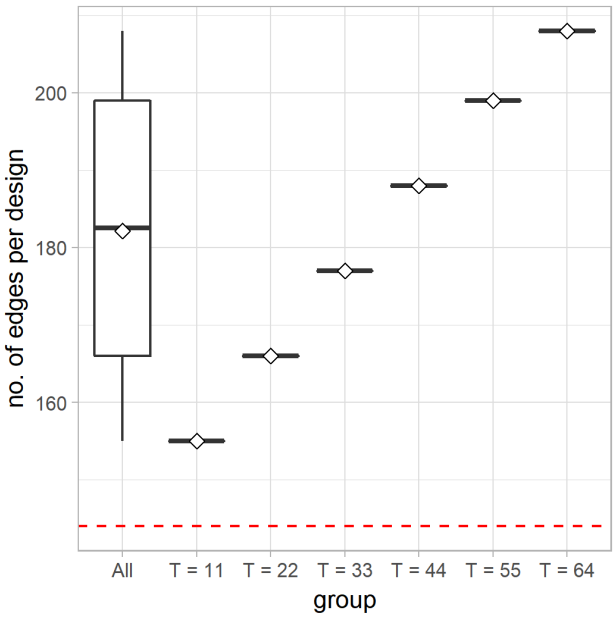

Figure 96. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

By adding an edge in an existing cell in a street graph, the operation of linking vertex to vertex splits the existing cell into two cells, therefore increasing the total number of cells by one. As shown in Figure 97, the total number of cells per design increases as the operation is applied more frequently. To be more specific, for a design generated by applying the operation T times on the initial 9 × 9 square-grid, we know that $|C| = 64 + T$.

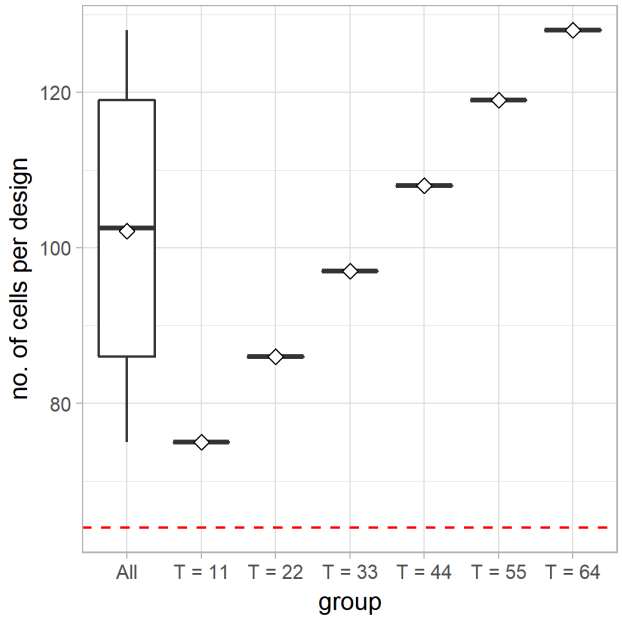

Figure 97. A boxplot (with mean diamonds) showing the distribution of the total number of cells per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the total number of cells in the initial 9 × 9 square-grid design (64).

(4) Vertex degree

By adding an edge between two vertices, the operation of linking vertex to vertex increases the degree for each of the vertices it links by one, respectively. As shown in Figure 98, based on all the vertices analyzed, the vertex degree ranges from 2 to 8. The most common vertex degree is 4. The mean vertex degree per design increases as the operation is applied more frequently. By the handshaking theorem, the sum of the degrees of the vertices of a graph is twice the number of its edges. Therefore, the mean vertex degree for a graph is $2|E| / |V|$. Therefore, for a design generated by applying the operation T times on the initial 9 × 9 square grid, the mean vertex degree for that design is $2(144 + T) / 81$, or $(288 + 2T) / 81$.

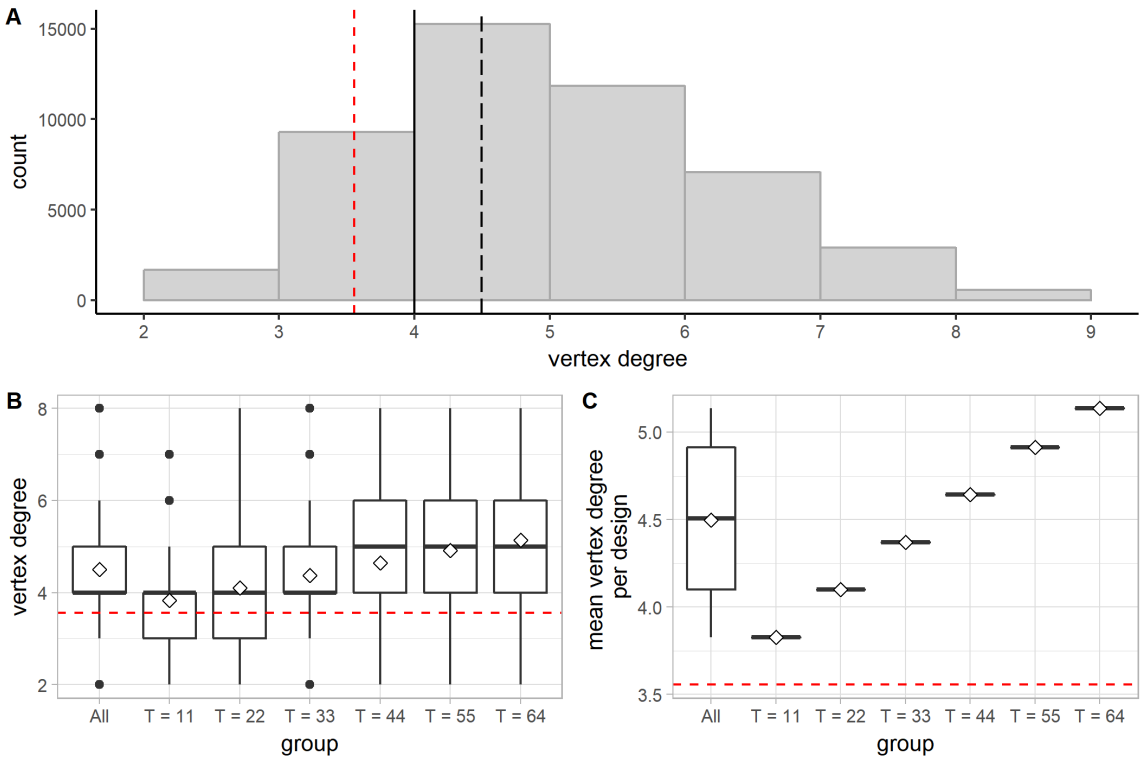

Figure 98. Distribution of the vertex degree (link vertex to vertex). A: A histogram showing the distribution of the vertex degree for all vertices in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design (≈ 3.556). B: A boxplot (with mean diamonds) showing the distribution of the vertex degree for all vertices and for vertices in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

By adding additional edges, the operation of linking vertex to vertex increases the total street length. As shown in Figure 99, the total street length per design increases as the operation is applied more frequently. Moreover, based on our algorithm and the initial setup of the street graph, the edges added during the generative process are of identical

length, namely the length of the diagonal of a 100 × 100 m square (≈ 141.42 m). Therefore, the total street length for a design generated by applying the operation T times on the initial 9 × 9 square-grid design is $14400 + \sqrt{100^2 + 100^2}T$, or roughly 14400 + 141.42T m.

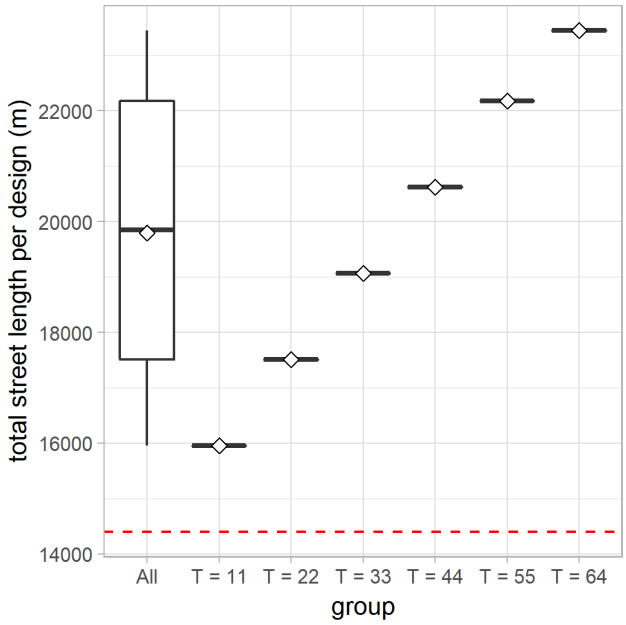

Figure 99. A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m).

(2) Total number of blocks per design

The total number of blocks is the same as the total number of cells per design.

(3) Total number of intersections per design

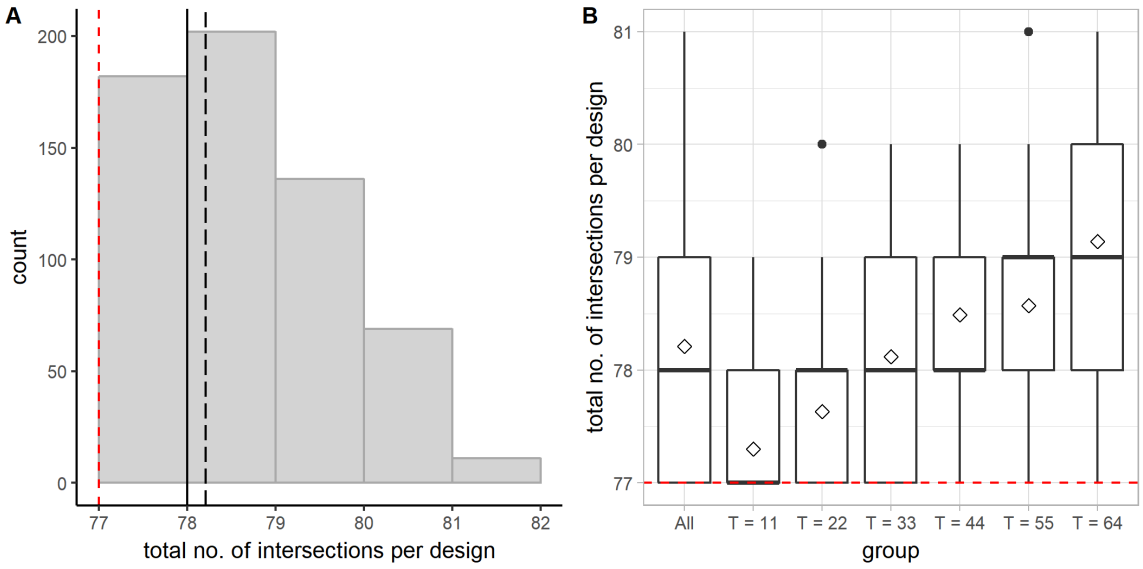

Figure 100. Distribution of the total number of intersections per design (link vertex to vertex). A: A histogram showing the distribution of the total number of intersections per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77). B: A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs.

By definition, the operation of linking vertex to vertex does not create new vertices. Therefore, the only way to change the number of intersections with this operation is to add degree to existing vertices that have degree lower than 3 so that they would be counted as intersections once their degree reaches 3. Since only the four vertices located at the corners of the initial square-grid graph have degree lower than 3, the total number of intersections can be increased at most by 4. This theoretical maximum is consistent with what we have observed in Figure 100. The total number of intersections per design ranges from 77 to 81. As the operation is applied more frequently, there is a greater chance that the vertices at the corners will be linked by

additional edges, thus we see an upward trend in the total number of vertices per design as *T* increases (Figure 100B).

(4) Distance between intersections

In the initial 9 × 9 square-grid design, the distance between intersections is uniformly 100 m except for the distance between the two T-intersections near each corner of the square grid which is 200 m. By linking vertices in the initial square grid, we create diagonals in the existing square blocks which are identical in length. The length of the diagonal path in a square block defines the distance between the two intersections it connects. Therefore, in the designs generated by our algorithm, the distance between intersections could take three different values: 100 m, 200 m, or $\sqrt{100^2 + 100^2}$ m. As shown in Figure 101B, the mean distance between intersections per design tends to increase as *T* increases. It is not surprising because as more vertices are linked with the diagonals, the distance between intersections is more frequently represented by the length of the diagonals, which is greater than the side length of the square block (100 m). (This finding may sound a bit counterintuitive: normally we would expect that by linking more vertices in the graph, we can surely make the graph more connected and also reduce the mean distance between intersections. However, it is not necessarily the case.)

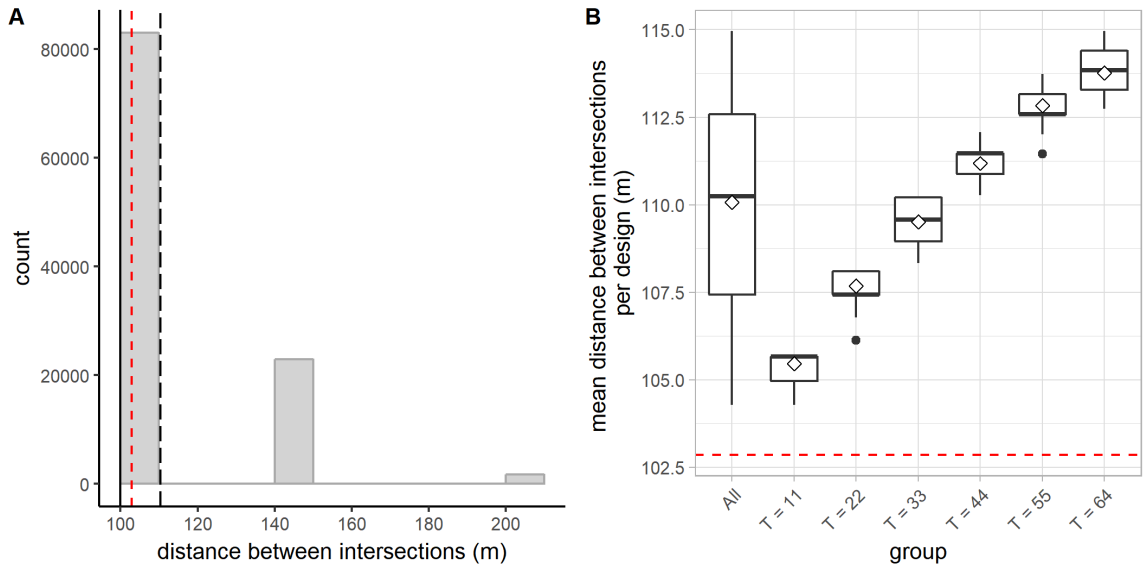

Figure 101. Distribution of the distance between intersections (link vertex to vertex). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Directional reach and directional distance

(1) DDL

The angle of deviation involved in transitioning between streets in the designs generated based on our algorithm can take only one of the four values—that is, 0°, 45°, 90°, or 135°. Therefore, as long as the threshold angle is set below 45°, no matter what specific value it is, the directional distance would stay the same. As shown in Figure 102, the distribution of the DDL values is not affected when the threshold angle is increased from 10° to 30°. However, to be consistent, the threshold angle is set to 20° in the following analysis.

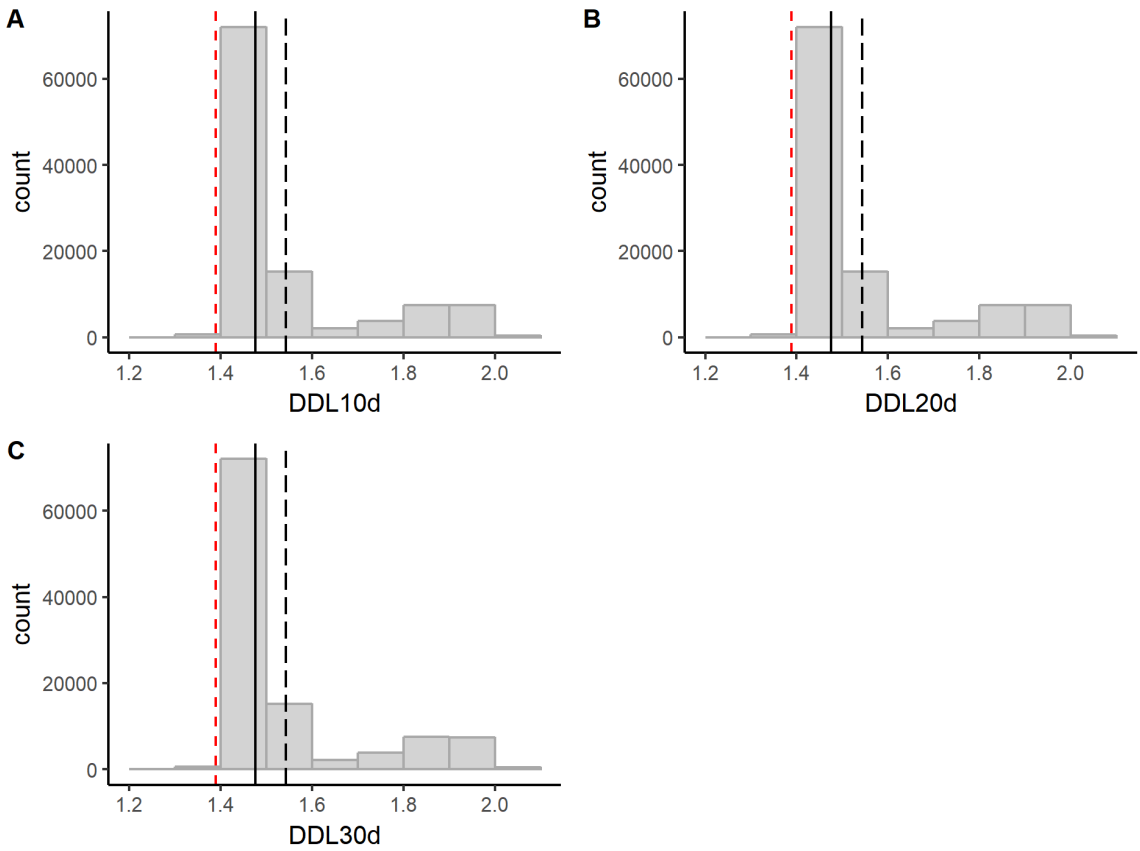

Figure 102. Distributions of DDL10d, DDL20d, and DDL30d (link vertex to vertex). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

As shown in Figure 102, in most cases, the DDL value assumed by a segment is between 1.4 and 1.5, close to the mean DDL20d of the initial 9 × 9 square-grid design (≈ 1.389). For all the segments analyzed, the DDL value ranges from 1.27 to 2.05. The extremely low DDL values observed are attributed to the formation of long diagonals across the entire superblock (so that it takes only one direction change to reach all the horizontally- and vertically-running streets). When the diagonals are not lined up, they tend to be more secluded than their orthogonal counterparts and therefore increase the mean directional distance. Since it is not very likely for the randomly placed diagonals to

align with each other and form very long lines across the grid, it is not very surprising to observe an overall upward trend in DDL20d (Figure 103). Note that this operation has a relatively small impact on the mean DDL20d per design, even if applied many times. As shown in Figure 103B, for all the designs analyzed, the mean DDL20d per design ranges between 1.44 and 1.65, a relatively small range.

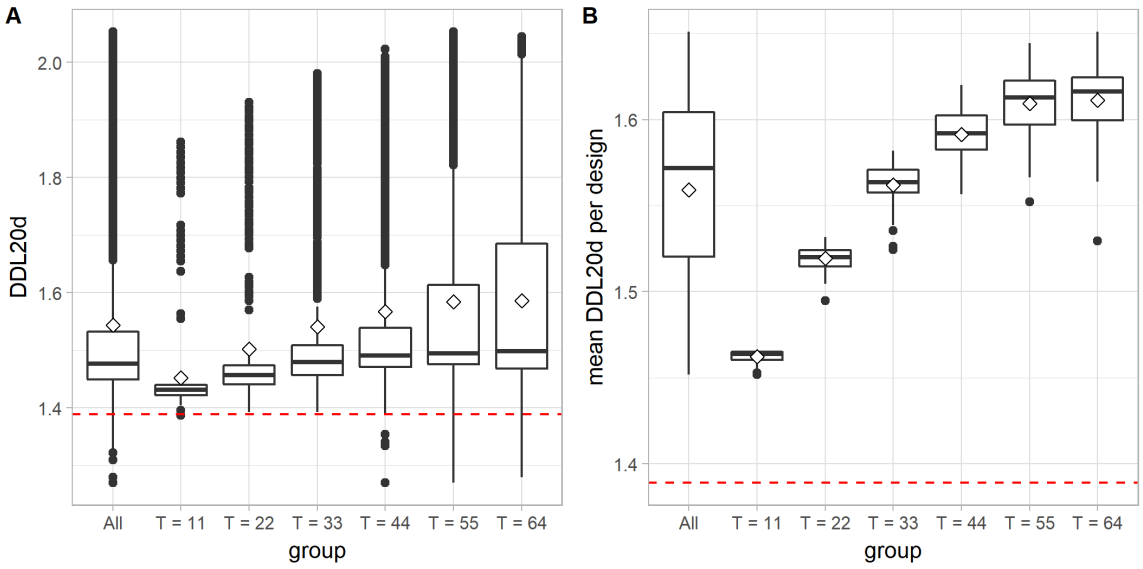

Figure 103. Distribution of DDL20d (link vertex to vertex). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 104, for most of the segments analyzed, their linear reach is 800 m. This is not surprising because most segments lie on the horizontally- and vertically-running streets and their linear reach is always 800 m. The linear reach can also take values that are integer multiples of the length of a diagonal. Theoretically, the

maximum linear reach can reach $8 \times \sqrt{100^2 + 100^2}$ m, or about 1131.37 m—that is, when the diagonals are lined up to form a "super" diagonal street cutting across the entire superblock. Such cases are indeed observed, as shown in Figure 104 A and B. However, it is not very likely for the randomly placed diagonals to align with each other and form very long lines across the grid. Therefore, the diagonals often have a smaller linear reach compared to their orthogonal counterparts. As shown in Figure 104C, the mean linear reach per design tends to decrease as *T* increases.

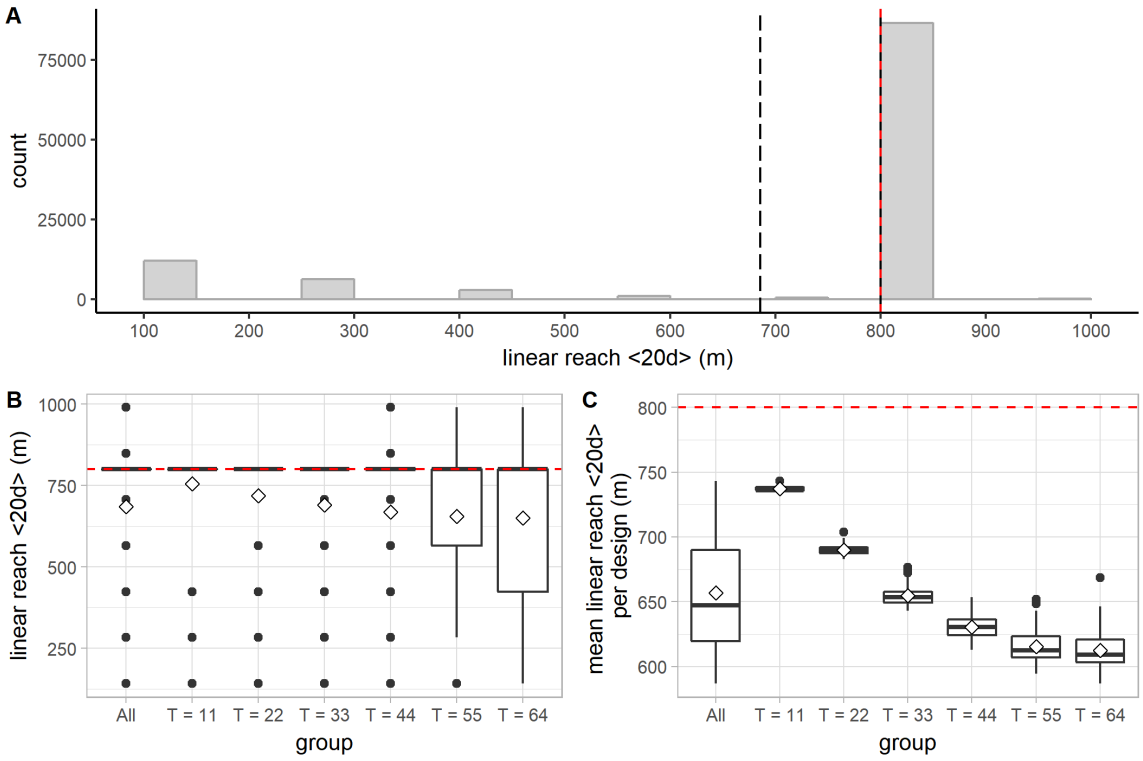

Figure 104. Distribution of the linear reach (link vertex to vertex). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

(3) 2-dc reach (dr2dc20d)

The distribution of the 2-dc reach as shown in Figure 105A resembles a comb distribution in which the bars are alternatively tall and short. It is easy to understand the distribution if we notice that for any design analyzed here, the 2-dc reach for the horizontal and vertical segments always equals the total street length in the design. To be more specific, for a design generated by applying the operation of linking vertex to vertex T times, the 2-dc reach for any horizontal or vertical segment in it equals $14400 + \sqrt{100^2 + 100^2}T$ m. Since the horizontal and vertical segments account for a large proportion of all the segments in each design, the counts corresponding to the bins they fall into are significantly higher than the others. On the other hand, although one can reach all the horizontally- and vertically-running streets within two direction changes from a diagonal street, it is not guaranteed that the rest of the diagonal streets in the design can be reached (these cases are fewer and correspond to the shorter bars in the histogram). Since the total street length per design increases as T increases, the mean 2-dc reach per design also tends to increase as T increases (Figure 105C).

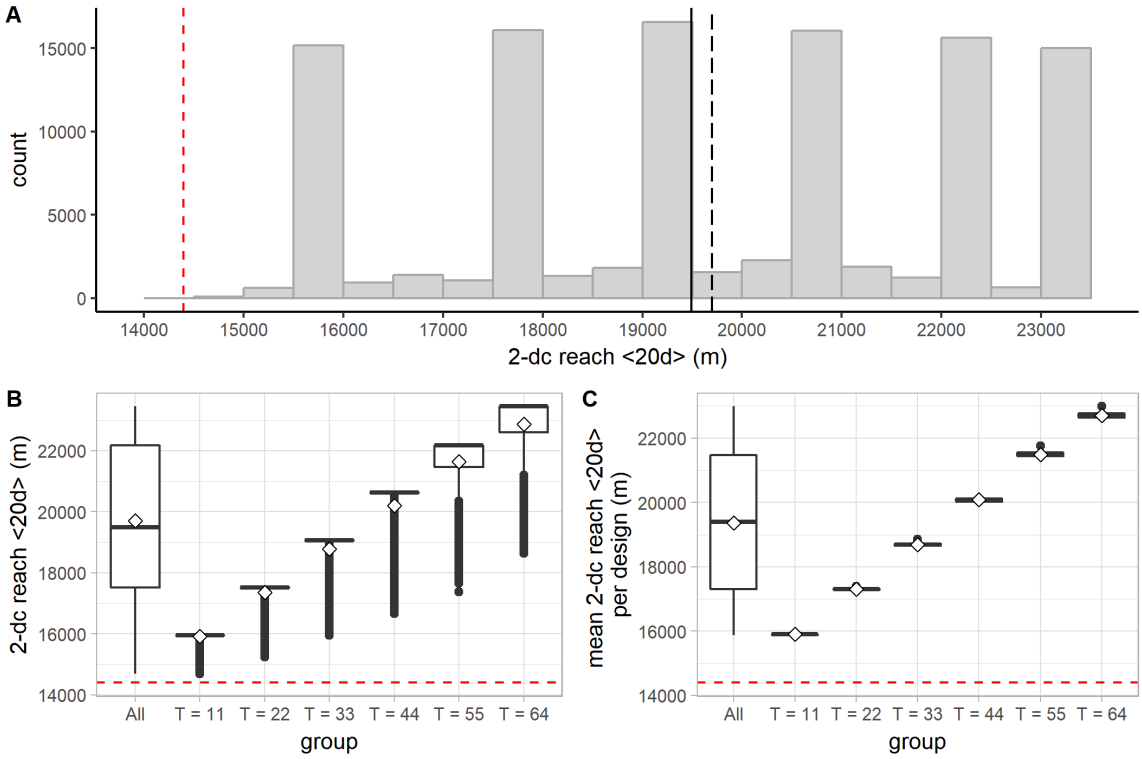

Figure 105. Distribution of the 2-dc reach(link vertex to vertex). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

There are in total 18 syntactic continuity lines in the initial 9 × 9 square-grid design, half running horizontally and half vertically. Each continuity line corresponds to a horizontal or a vertical street, and comprises eight segments, therefore the fragmentality for the square-grid design is identical to the fragmentality for an individual street in the

square-grid design, which is 1/8, or 0.125. Applying the operation of linking vertex to vertex adds diagonals to the initial design, and in doing so, we tend to increase the fragmentality for the whole design because of the diagonals, even when lined up to form the "super" diagonal across the entire grid, can never yield a fragmentality value lower than 1/8. As shown in Figure 106, when analyzed in groups, the fragmentality per design tends to increase as *T* increases from 11 to 55.

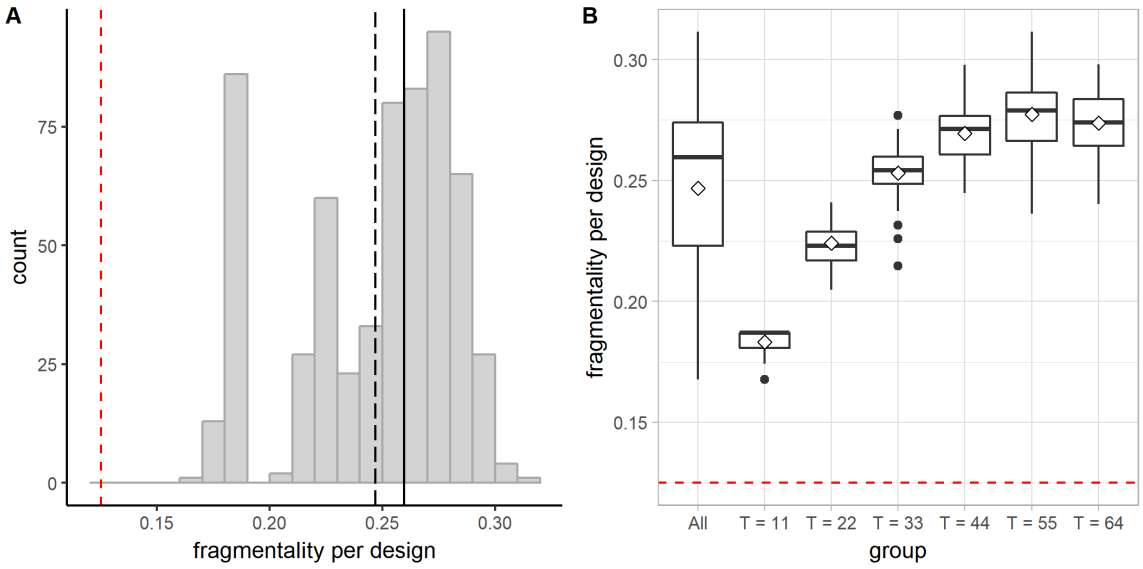

Figure 106. Distribution of the fragmentality per design (link vertex to vertex). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

Based on our algorithm and the setup of the initial street graph, a single application of the operation of linking vertex to vertex would split an existing square block in half—that is, by adding diagonals, we also subdivide blocks. Therefore, the

more frequently we apply the operation, the more small and triangular blocks we get, and the lower the mean block area per design gets, as shown in Figure 107. To be more specific, for a design generated by applying the operation of linking vertex to vertex T times on the initial 9 × 9 square-grid design, its mean block area equals 800 × 800 / (64 + T) m², or 640000 / (64 + T) m².

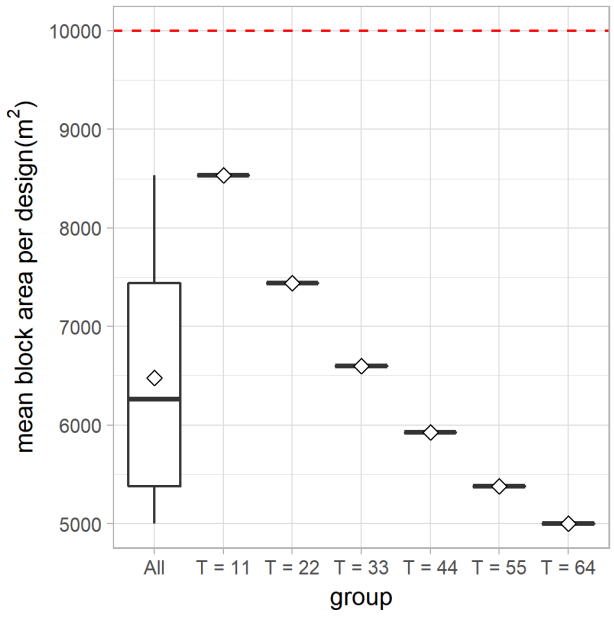

Figure 107. A boxplot showing the distribution of the mean block area per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²).

The areas of blocks inside a design become more varied initially, but soon, as the operation is applied more and more frequently, more and more blocks in the design become triangular blocks of the same shape and size, eventually eliminating any variation in block size (Figure 108).

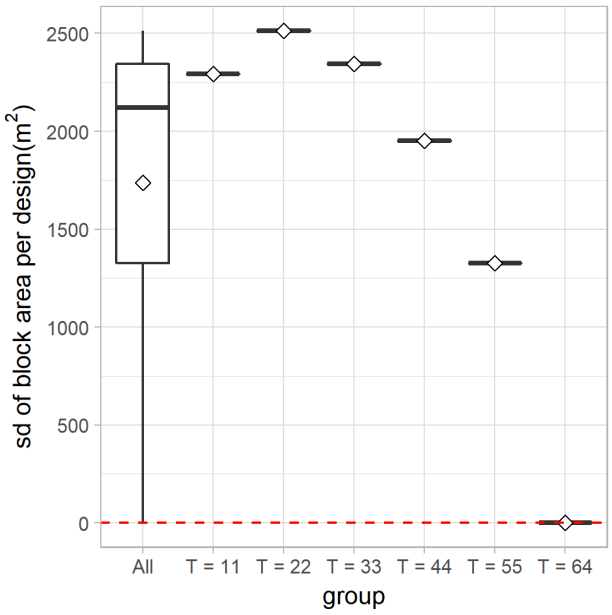

Figure 108. A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²).

(3) Block perimeter

The more frequently we apply the operation, the greater the proportion of small triangular blocks we get. Since the perimeter of the triangular block resulted from the operation is smaller than the perimeter of the original square block, the mean block perimeter per design decreases as the operation is applied more frequently, as shown in Figure 109. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, its mean block perimeter equals

$\frac{(14400\times2-800\times4+2\sqrt{100^2+100^2}T)}{64+T}$ m, or $\frac{(25600+200\sqrt{2}T)}{64+T}$ m.

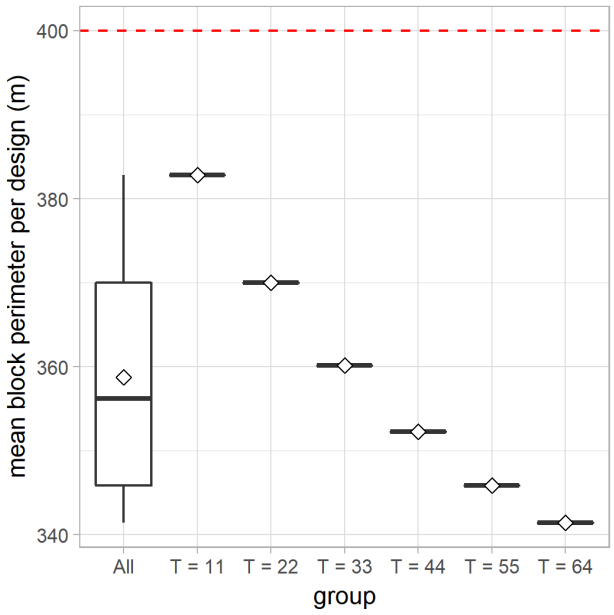

Figure 109. A boxplot showing the distribution of the mean block perimeter per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m).

The perimeters of blocks inside a design become more varied initially, but soon, as the operation is applied more and more frequently, more and more blocks in the design become triangular blocks of the same shape and size, eventually eliminating any variation in the length of block perimeter (Figure 110).

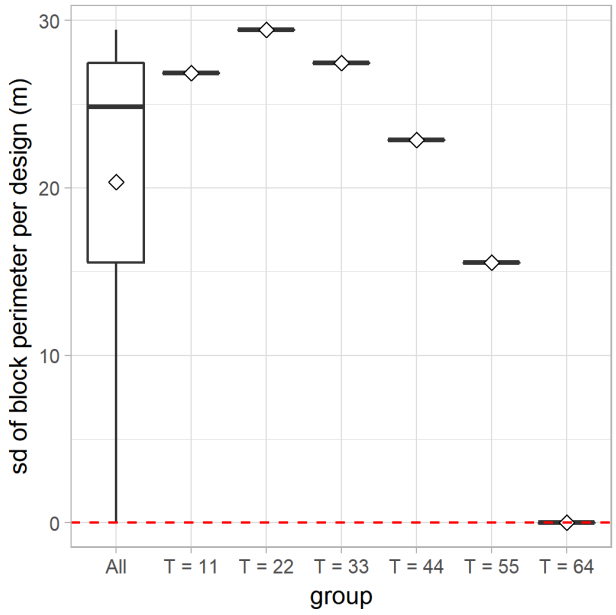

Figure 110. A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m).

(4) Standardized block area-perimeter ratio (SAPR)

The standardized block area-perimeter ratio for the triangular blocks resulted from this operation is $2\sqrt{2}-2$, or about 0.83. After applying the operation of linking vertex to vertex T times on the initial 9 × 9 square-grid design, we would have 64 – T square blocks and 2T triangular blocks in the design. Therefore, the mean SAPR of the design generated by applying the operation T times on the initial square-grid design equals $[(64-T)+2T(2\sqrt{2}-2)]/(64+T)$. It follows that the mean SAPR per design decreases as T increases, as shown in Figure 111.

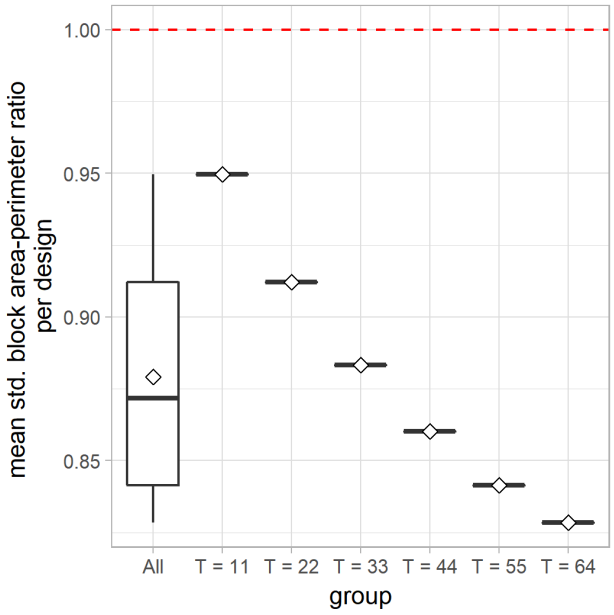

Figure 111. A boxplot showing the distribution of the mean standardized block area-perimeter ratio per design for all designs and for each group of designs (link vertex to vertex). The red dashed line indicates the mean standardized block area-perimeter ratio for the initial 9 × 9 square-grid design (1.00).

Diversity in syntactic conditions

(1) Total number of and the proportion of distinct DDL20d values per design

The total number of distinct DDL20d values per design ranges from 7 to 55. As shown in Figure 112A, in more than half cases, there are 30 or more distinct DDL20d values assumed by the segments in each design. When analyzed in groups, as *T* increases, the total number of distinct DDL20d values per design tends to increase. The proportion of distinct DDL20d values per design ranges from 0.05 to 0.26. As shown in Figure 112C, when analyzed in groups, the proportion of distinct DDL20d values per design tends to increase as *T* increases. However, the range covered by the proportion is

relatively small and there is generally a lack of diversity in terms of the syntactic condition.

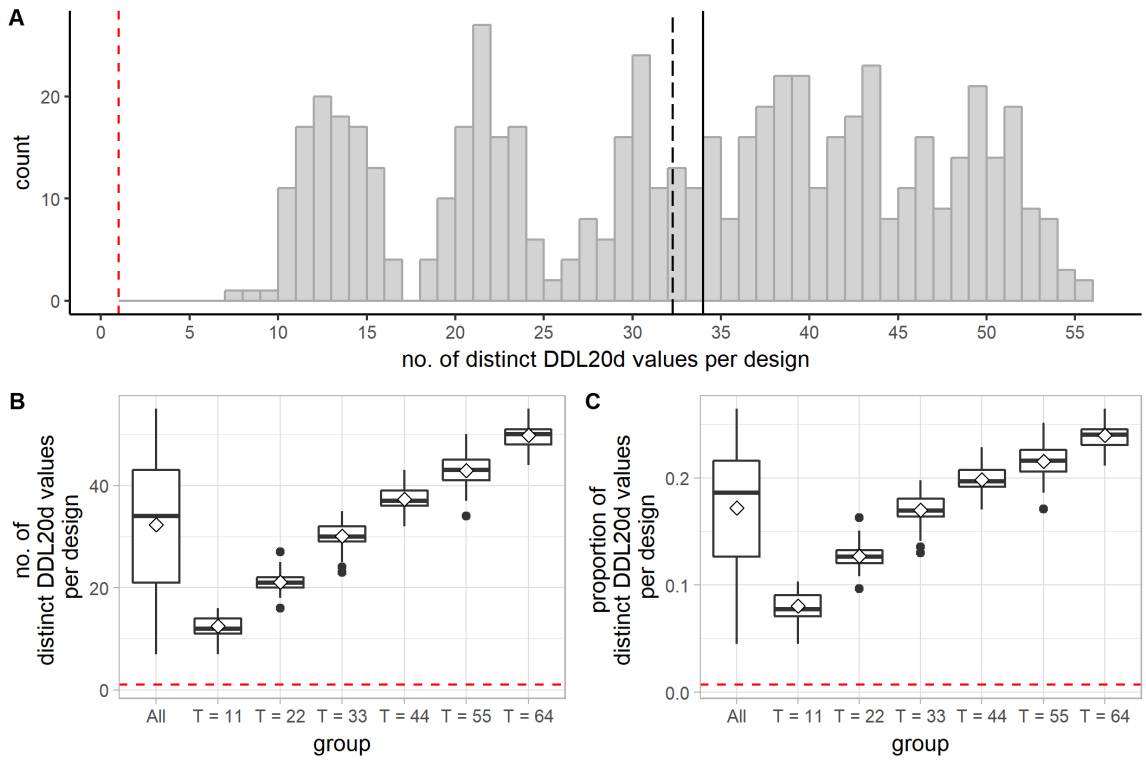

Figure 112. Distribution of the total number and the proportion of distinct DDL20d values per design (link vertex to vertex). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d values per design

The standard deviation of DDL20d values per design ranges from 0.10 to 0.20. In more than 60% of the cases, the standard deviation of DDL20d values is greater than

0.16. As shown in Figure 113B, when analyzed in groups, as *T* increases, the standard deviation of DDL20d values per design tends to increase as well (although the upward trend becomes less pronounced after a certain point).

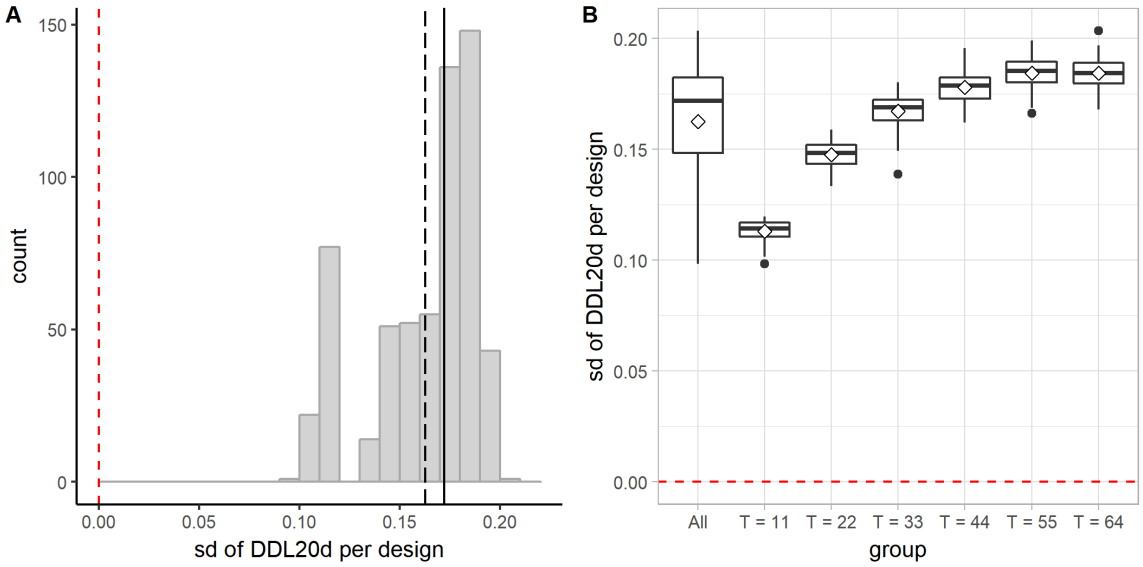

Figure 113. Distribution of the standard deviation of DDL20d per design (link vertex to vertex). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

6.4 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: This operation maintains the total number of vertices per design. Every single application of the operation increases the total number of edges by one and increases the total number of cells by one as well.

The operation increases the degree of the vertices it operates on, therefore the mean vertex degree per design increases as the operation is applied more frequently. Based on our algorithm, the vertex degree can never exceed 8.

The impact of the operation on the density of a street network: This operation increases the total street length and the total number of blocks per design. In these regards, it makes the street network denser. However, this operation tends to increase the mean distance between intersections per design because the distance between intersections is increasingly represented by the length of the diagonal streets cutting through the square blocks.

The impact of the operation on the directional reach/distance: The operation could create diagonal streets that are more integrated (has a lower directional distance per length) than the horizontal and vertical streets. However, more often it creates short diagonal streets which are more secluded than the horizontal/vertical ones. Therefore, it tends to increase the mean DDL20d for a design. Interestingly, this operation tends to decrease the mean linear reach per design while increasing the 2-dc reach per design. The former trend is attributed to the short diagonals added to the street network, while the latter is attributed to the increasing total street length per design.

The impact of the operation on the regularity of a street network: This operation tends to increase the fragmentality per design. The mean block area decreases as the total number of blocks per design increases because of the operation. The mean block perimeter also decreases as the operation is applied more frequently. Analysis of the standard deviation of the mean block area per design and the mean block perimeter per

design shows that both the block area and block perimeter get differentiated initially but get identical again when all the square blocks in the initial square-grid design are split into triangular blocks of the same shape and size. As the operation converts square blocks to triangular blocks, it also decreases the mean standardized block area-perimeter ratio per design.

The impact of the operation on the diversity of syntactic conditions: This operation tends to increase the diversity of the syntactic condition in a superblock design, as evidenced by the increasing proportion of distinct DDL20d values per design and the increasing standard deviation of DDL values per design. However, there is in general a lack of diversity in the syntactic condition in designs generated by this operation because the proportion of the distinct DDL values is relatively low.

CHAPTER 7
DISJOINING VERTICES: CONSOLIDATING URBAN BLOCKS

7.1 Operation: Disjoin Vertices

7.1.1 Definition of operation

The operation of disjoining vertices removes an edge and merges the two cells that shared the edge (Figure 114).

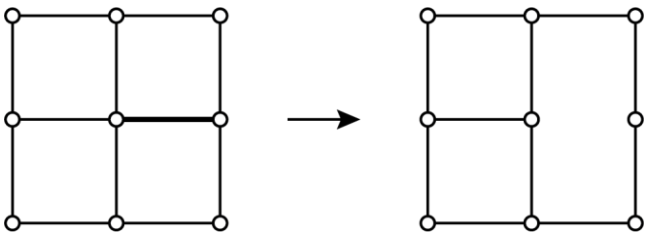

Figure 114. Disjoin two vertices.

7.1.2 Parameters

Once the edge to be removed is specified, there is no ambiguity about how to perform the operation. Therefore, no parameters are needed.

7.2 Generative Algorithm

7.2.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock design, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by X;
- the number of streets running horizontally in the initial regular grid, denoted by Y;
- the edge to be removed, denoted by e;
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled by the following parameters: l_x, l_y, X, Y, e, T.

7.2.2 General description

We start from a street graph that represents a regular grid. Then we randomly pick an edge, e, and remove e from the street graph. We continue to do so until the operation has been successfully performed a specified number of times. The operation can only be successfully performed if *all* of the following conditions are met.

1. The edge to be removed does not lie on the border of the street graph.
2. The edge to be removed is not a dangling edge.
3. The edge to be removed is not a bridge.

The last condition is included to prevent the creation of disconnected graphs. Below illustrate a few situations where one of the above conditions is not met.

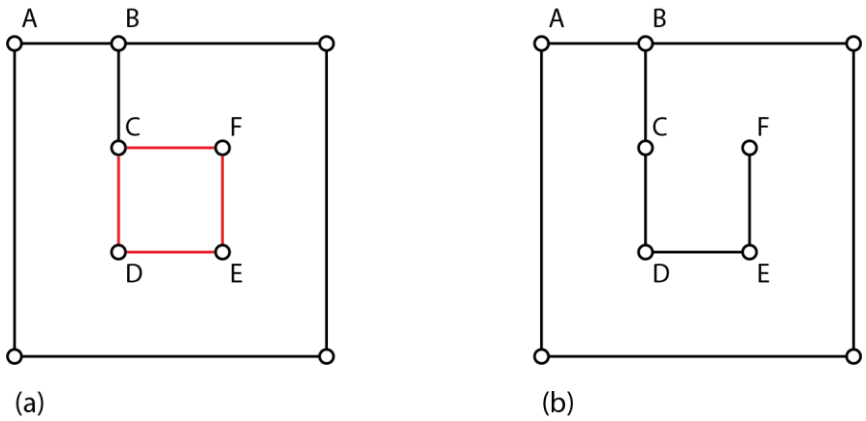

Figure 115. Examples of illegitimate moves (disjoin vertices).

In the example shown in Figure 115a, there are two cells in the street graph. (The boundary of the smaller cell in the center is highlighted in red.) While any one of the red edges can be removed, none of the black edges can. For example, the edge {*A*, *B*} cannot be removed because it lies on the border of the street graph—removing it would lead to a violation of Condition 1; the edge {*B*, *C*} cannot be removed because it is a bridge and removing it would make the graph disconnected—that is, a violation of Condition 3. If we decide to remove one of the red edges, say the edge {*C*, *F*}, then there is only one cell left, as shown in Figure 115b. At this point, none of the edges can be further removed. The edges {*C*, *D*} and {*D*, *E*} cannot be removed because they are now bridges and removing any one of them would disconnect the graph. The edge {*E*, *F*} cannot be removed either because it is a dangling edge—removing it would lead to a violation of Condition 2.

7.2.3 Pseudocode

Suppose that we have already generated the initial street graph G which represents a regular grid that is l_x units long, l_y units wide, with X number of streets running vertically and Y number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of disjoining vertices based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random_choice* (*s*) refers to a procedure which returns a random element from the set *s*.

ALGORITHM 7.1: Disjoin vertices

```
procedure disjoin_vertices (G, T)
counter ≔ 0
while counter < T
        e ≔ random_choice (E)
        if e does not lie on the border of G
        and e is not a dangling edge
        and e is not a bridge then
                remove e
                counter ≔ counter + 1
```

7.3 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the different groups of designs, we varied only one control parameter—namely T, the total number of operations to be performed—and kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9.

7.3.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation on the initial 9 × 9 square-grid design—that is, the values of T. It is easy to see that, based on our algorithm, each time we disjoin two vertices, two internal blocks are merged into one big block. Since there are in total 64 internal blocks in the initial 9 × 9 square-grid design, we can at most perform the operation of disjoining vertices 63 times, after which there will be no more internal blocks to be merged with and the superblock will be filled with cul-de-sacs. Therefore, we set the largest T value, T_{max}, to 63. The minimum number of times to apply the operation, T_{min}, was subsequently determined by finding the nearest integer for 63 / 6, which is 11. Then we simply set T to integer multiples of 11 to generate the other groups of designs.

Examples of designs from each group are shown in Figure 116.

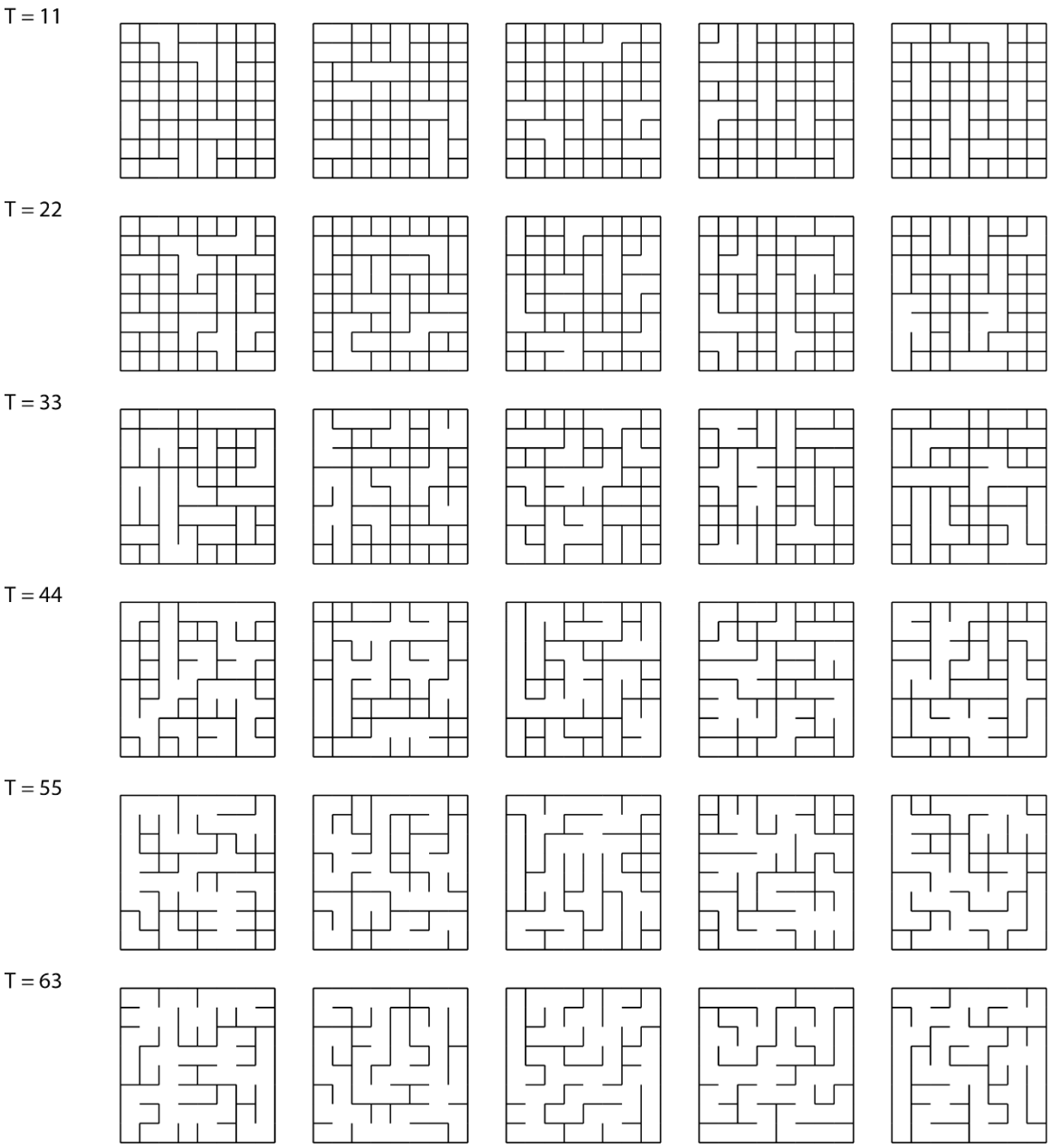

Figure 116. Examples of designs from each group (disjoin vertices).

7.3.2 Data analysis

Six groups of designs were generated by applying the operation of disjoining vertices 11, 22, 33, 44, 55, and 63 times, respectively, on the initial 9 × 9 square-grid design. Each group consists of 100 designs. The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of designs.

Elementary graph properties

(1) Number of vertices

Based on our algorithm, the operation only removes edges but never removes vertices. Therefore, the total number of vertices per design always stays the same—that is, $|V| = 81$.

(2) Number of edges

By definition, each application of the operation removes one edge, thus reducing the total number of edges by one. As shown in Figure 117, the total number of edges per design decreases as the operation is applied more frequently. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, we know that $|E| = 144 - T$.

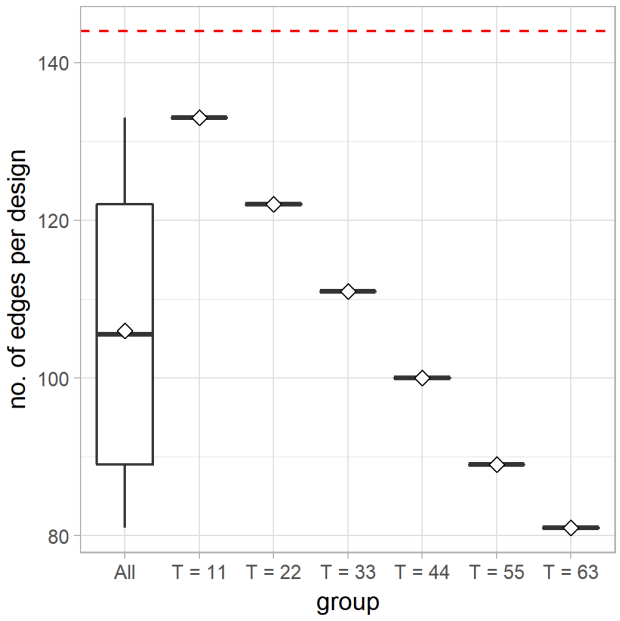

Figure 117. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (disjoin vertices). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

Based on our algorithm, each time we apply the operation, we remove an edge and subsequently merge the two adjacent cells that contained the edge just removed, thus reducing the total number of cells by one. As shown in Figure 118, the total number of cells per design decreases as the operation is applied more frequently. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, we know that $|C| = 64 - T$.

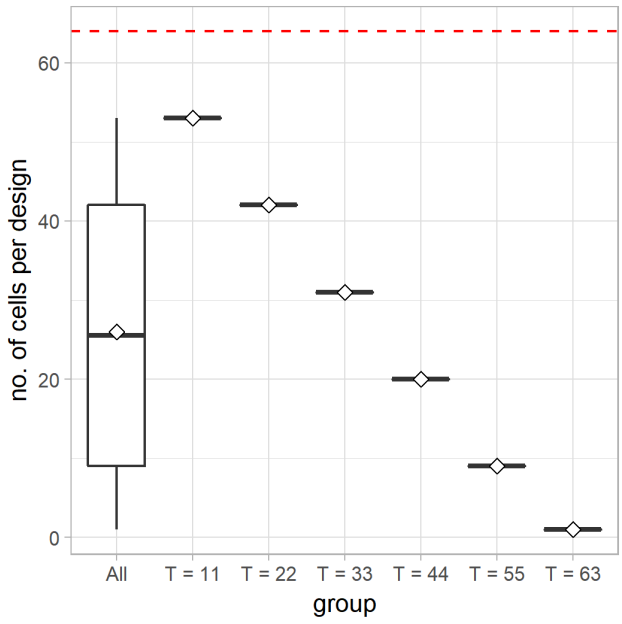

Figure 118. A boxplot (with mean diamonds) showing the distribution of the total number of cells per design for all designs and for each group of designs (disjoin vertices). The red dashed line indicates the total number of cells in the initial 9 × 9 square-grid design (64).

(4) Vertex degree

By definition, the operation of disjoining vertices reduces the number of incident edges for the affected vertices, thus reducing the degree for those vertices. The vertex degree for all the vertices analyzed ranges from 1 to 4. The presence of vertices of degree one suggests the presence of cul-de-sacs, or dead-end streets. The most common vertex degree is 3, suggesting the prevalence of T-intersections. By the handshaking theorem, the sum of the degrees of the vertices of a graph is twice the number of its edges. Therefore, the mean vertex degree for a graph is $2|E| / |V|$. We know that for a design generated by applying the operation of disjoining vertices T times on the initial square-grid design, $|V| = 81$ and $|E| = 144 - T$, therefore, the mean vertex degree for the design is

2(144 – *T*) / 81. As shown in Figure 119B, the mean vertex degree per design decreases as the operation is applied more frequently.

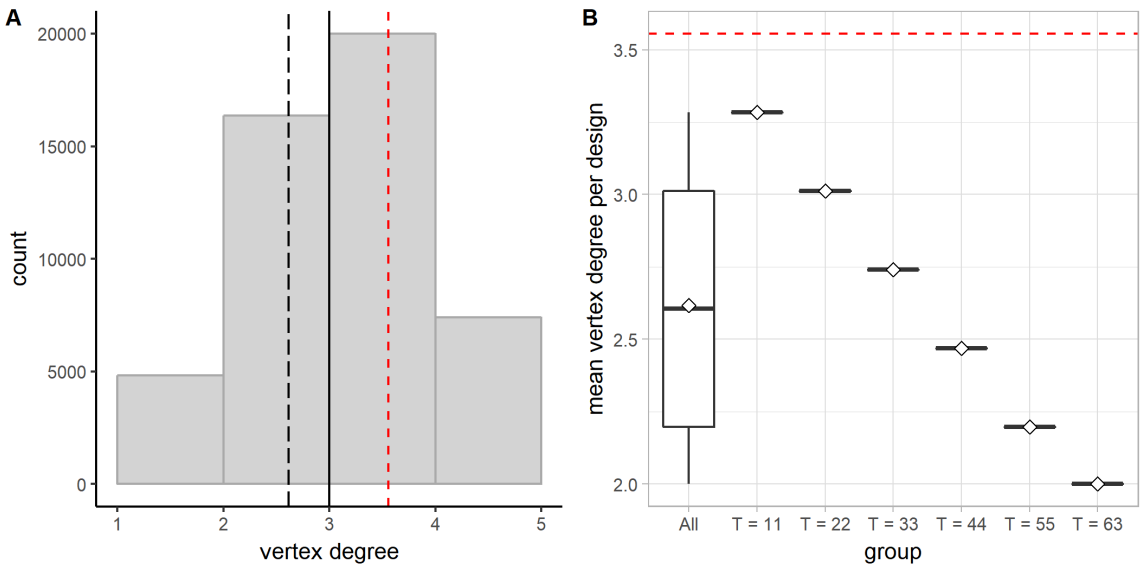

Figure 119. Distribution of the vertex degree (disjoin vertices). A: A histogram showing the distribution of the vertex degree for all vertices in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In both subfigures, the red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design (≈ 3.556). B: A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

By definition, the operation of disjoining vertices removes existing edges in a street graph, thus reducing the total street length. As shown in Figure 120, the total street length per design decreases as the operation is applied more frequently. More specifically, the total street length for a design generated by applying the operation of disjoining vertices *T* times on the initial 9 × 9 square-grid design is 14400 – 100*T* m.

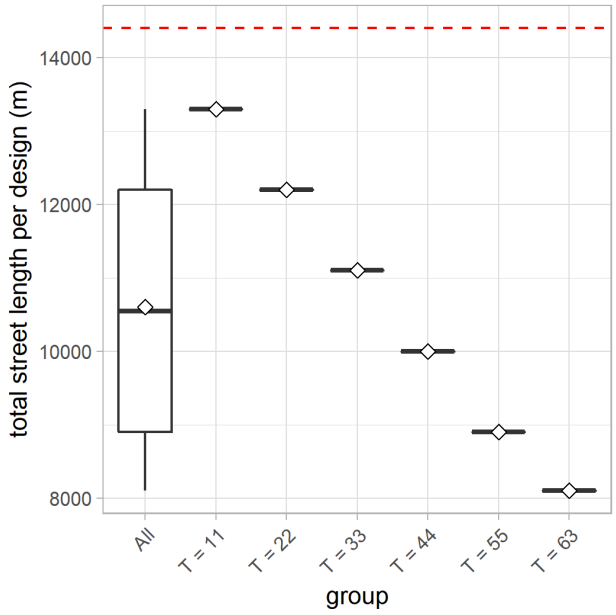

Figure 120. A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs (disjoin vertices). The red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m).

(2) Total number of blocks per design

The total number of blocks is the same as the total number of cells per design.

(3) Total number of intersections per design

By reducing the degree of existing vertices, the operation of disjoining vertices tends to reduce the number of intersections (because once the vertex has degree lower than 3, it would not be counted as an intersection anymore). As shown in Figure 121, the total number of intersections per design ranges from 15 to 75. When analyzed in groups, as *T* increases, the total number of intersections per design tends to decrease.

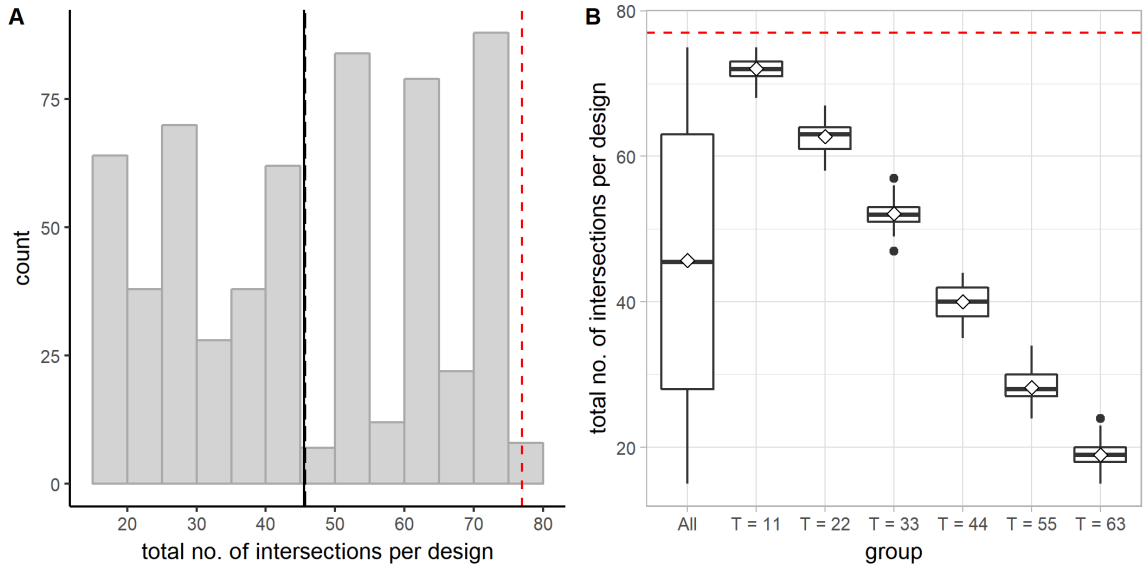

Figure 121. Distribution of the total number of intersections per design (disjoin vertices). A: A histogram showing the distribution of the total number of intersections per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77). B: A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs.

(4) Distance between intersections

As shown in Figure 122, while in most cases the distance between intersections is 100 m, in extreme cases the distance between intersections could reach 1200 m—that is more than a third of the entire perimeter of the 800 m × 800 m superblock (3200 m). As the total number of intersections decreases, the distance between intersections tends to be longer. The extremely long distance between intersections tends to occur in the groups of designs generated by more frequent application of the operation of disjoining vertices (Figure 122B). As shown in Figure 122C, when analyzed in groups, as *T* increases, the distance between intersections per design tends to not only increase but cover a greater range of values as well. The rate of increase also tends to be faster as *T* increases.

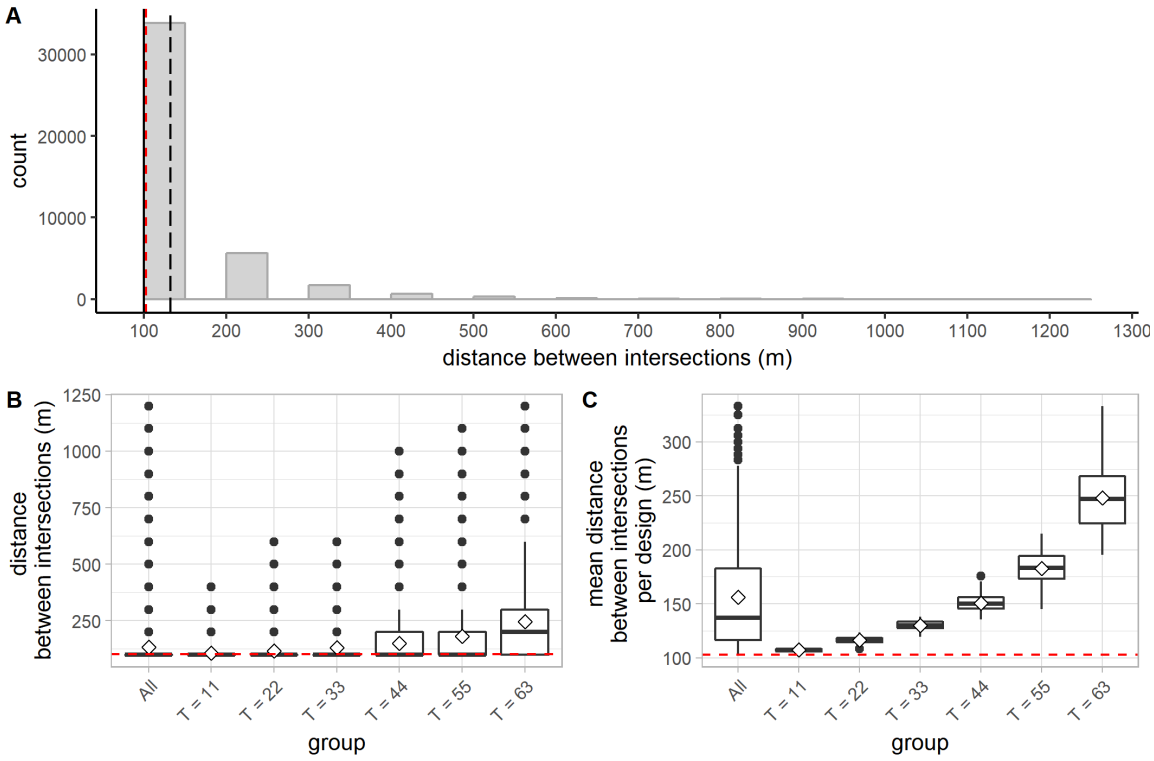

Figure 122. Distribution of the distance between intersections (disjoin vertices). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the distance between intersections observed in all designs and in each group of designs. C: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Directional reach and directional distance

(1) DDL

Since the angle of deviation involved in transitioning between segments in the designs analyzed here can only take two values, 0° or 90°, it does not matter what threshold angle we use to analyze the directional distance as long as it is set below 90°. As shown in Figure 123, the distribution of the DDL values is not affected when the

threshold angle is increased from 10° to 30°. To be consistent, we set the threshold angle to 20° in the following analysis.

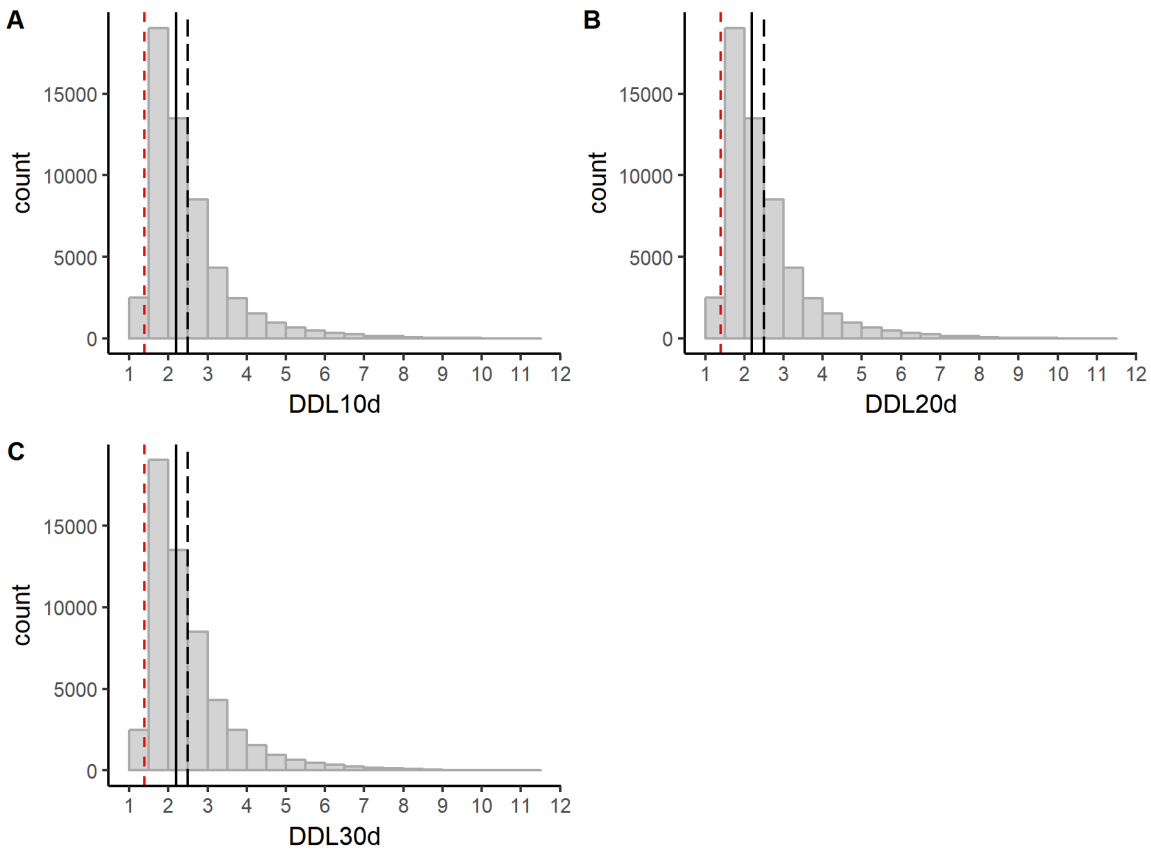

Figure 123. Distributions of DDL10d, DDL20d, and DDL30d (disjoin vertices). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

As shown in Figure 123, while more than half of all the segments assume a DDL value that is lower than 2.20, the DDL value assumed by the most segregated segment can be greater than 10—that is, on average, everything is more than 10 turns away from that segment! As shown in Figure 124, there is generally an upward trend in DDL20d as the operation is applied more frequently. The operation is particularly effective in

creating extremely highly segregated segments in the design. When analyzed in groups, as *T* increases, the mean DDL20d per design also tends to increase and generally assumes an increasingly wider range of values (Figure 124B).

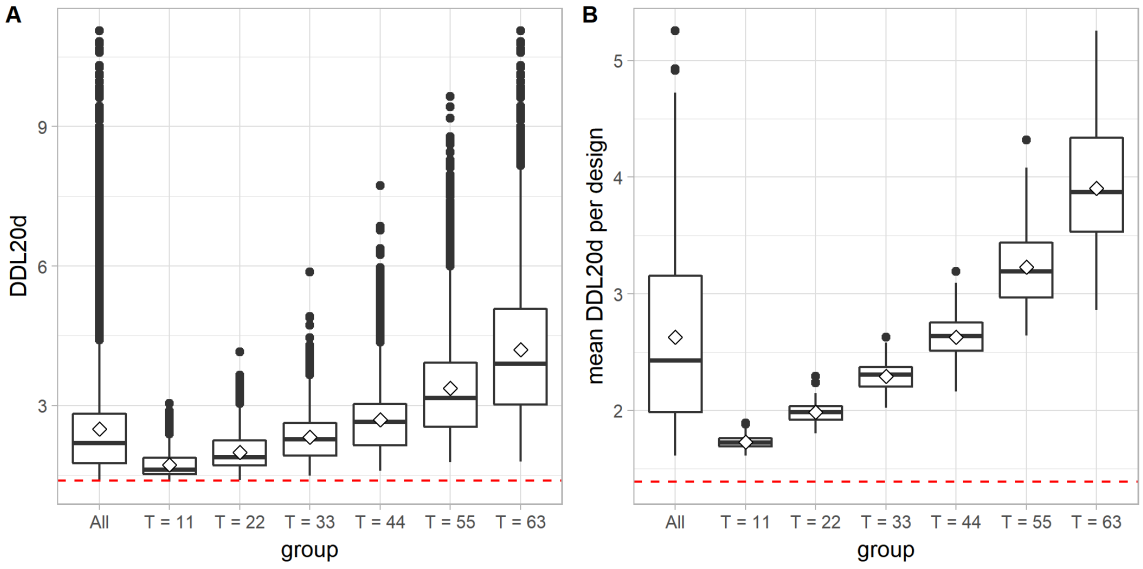

Figure 124. Distribution of DDL20d (disjoin vertices). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 125, for all the segments analyzed, the linear reach ranges from 100 to 800 m. Because of the initial setup of the street graph and the nature of the operation, the observed values of linear reach are all integer multiples of 100 m. As shown in Figure 125B, the low linear reach values occur more often as the operation is applied more frequently. When analyzed in groups, as *T* increases, the mean linear reach per design decreases (Figure 125C).

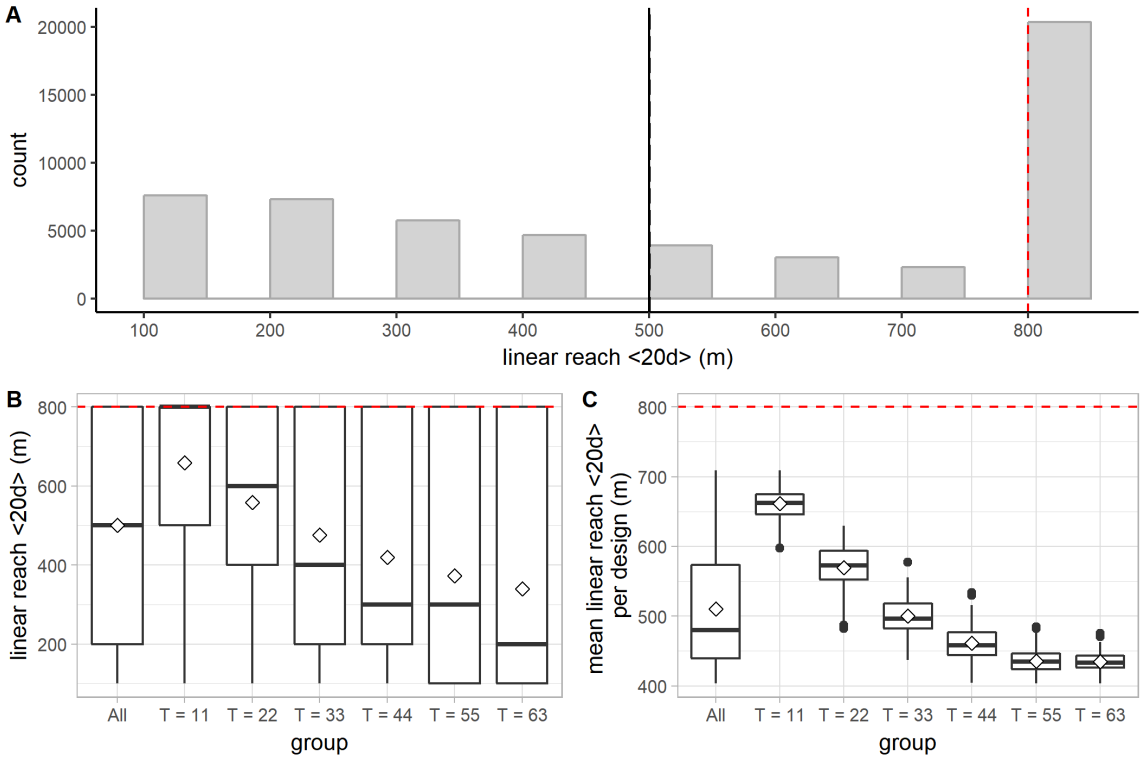

Figure 125. Distribution of the linear reach (disjoin vertices). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

(3) 2-dc reach (dr2dc20d)

For all the segments analyzed, the 2-dc reach ranges from 300 to 13200 m. As shown in Figure 126, the 2-dc reach values distribute more or less evenly across the whole range. As more vertices are disjoined, it becomes more difficult to reach other edges without taking many turns. As shown in Figure 126B, there a downward trend for the 2-dc reach as the operation is applied more frequently. When analyzed in groups, as *T* increases, the mean 2-dc reach per design tends to decrease as well (Figure 126C).

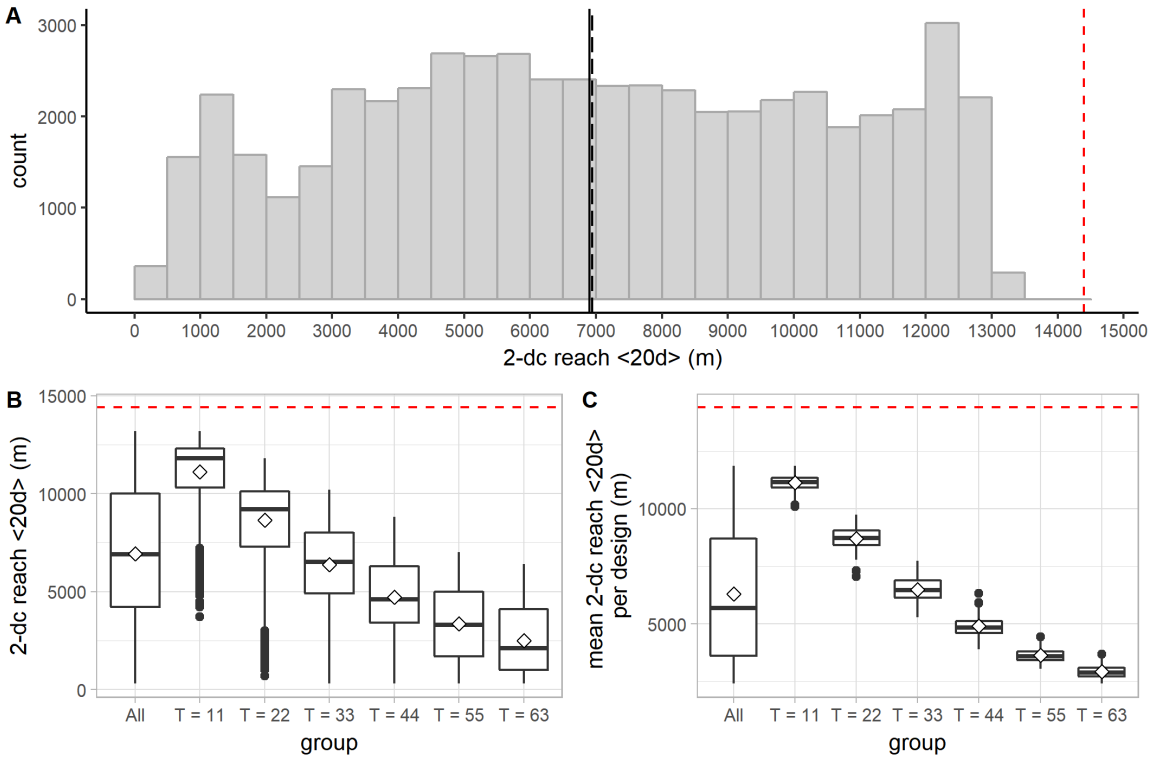

Figure 126. Distribution of the 2-dc reach (disjoin vertices). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

By removing edges, the operation of disjoining vertices disrupts linear extensions of streets in the initial square-grid design, therefore increasing the fragmentality per design. As shown in Figure 127, the maximum fragmentality per design can be greater than 0.7. When analyzed in groups, as *T* increases, the fragmentality per design clearly

tends to increase (Figure 127B). Notice that the maximum fragmentality observed in the group with $T = 11$ is 0.22, while the minimum fragmentality observed in the group with $T = 63$ is 0.55. In other words, the most regular design in the group with $T = 63$ is still 2.5 times as "fragmented" as the most irregular design in the group with $T = 10$.

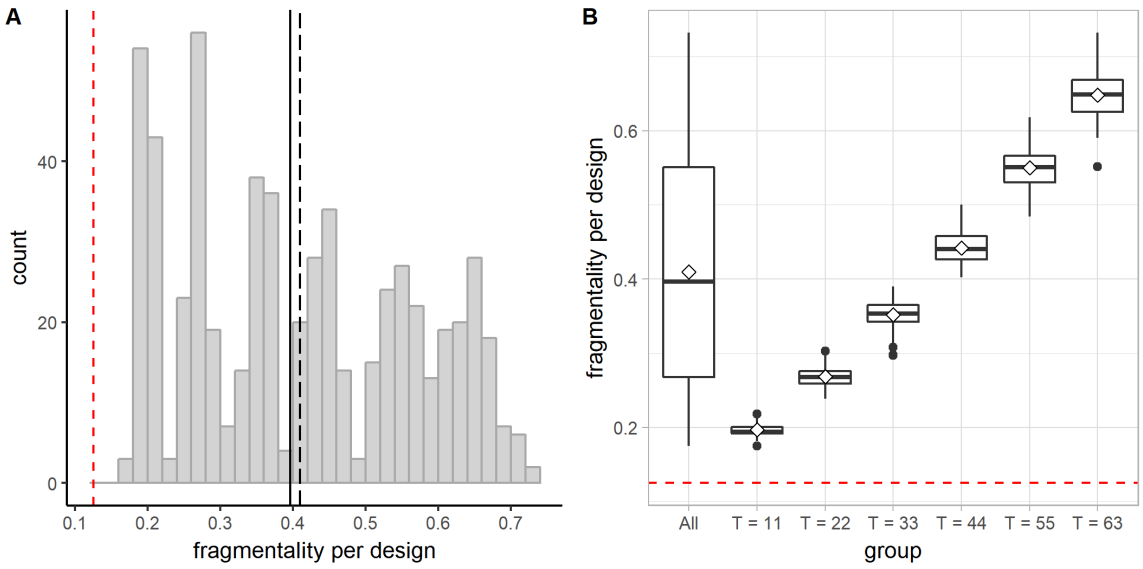

Figure 127. Distribution of the fragmentality per design (disjoin vertices). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

Since the operation of disjoining vertices merges two existing blocks into one, it tends to create large blocks. As shown in Figure 128, for all the blocks analyzed, the block area ranges from 10000 up to 640000 m²—a wildly broad range of values. When analyzed in groups, the maximum block area observed becomes increasingly larger as T increases (Figure 128B). As T increases, the mean block area per design increases as well

(Figure 128C). More specifically, the mean block area for a design generated by applying the operation of disjoining vertices T times on the initial 9 × 9 square-grid design is 640000 / (64 – T) m².

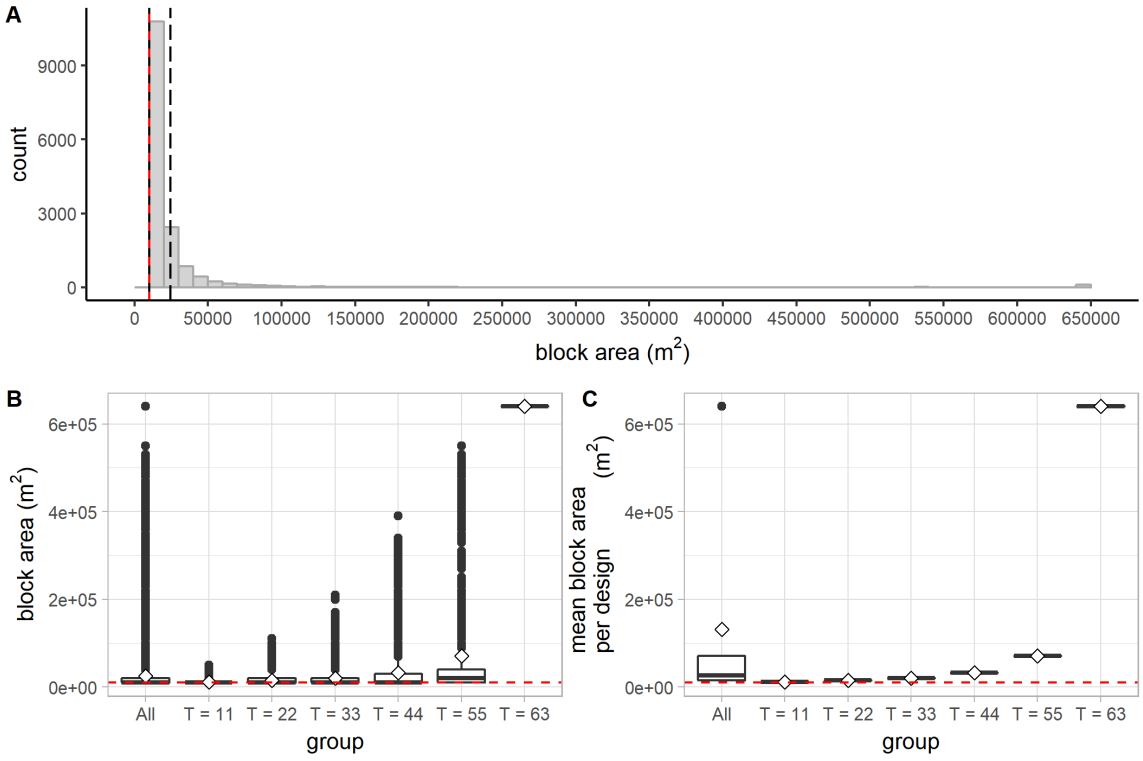

Figure 128. Distribution of the block area (disjoin vertices). A: A histogram showing the distribution of the area for all the blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²). B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs. C: A boxplot showing the distribution of the mean block area per design for all designs and for each group of designs.

As the operation is applied more frequently, the block area in a design also becomes more varied. As shown in Figure 129, the standard deviation of the block area per design tends to increase as the operation is applied more frequently.

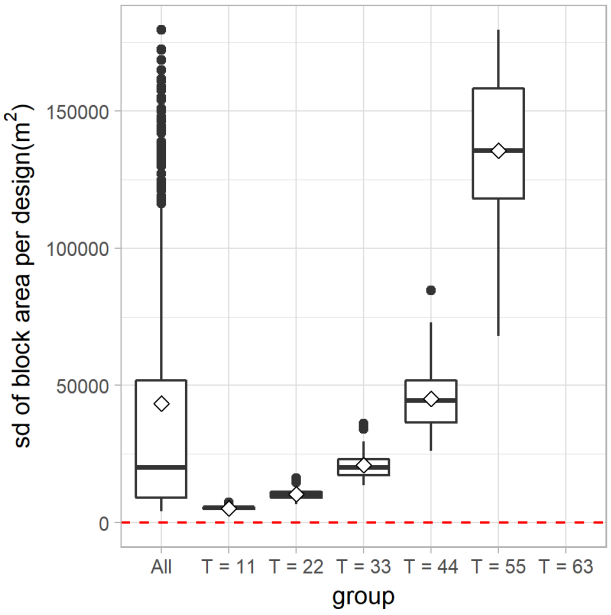

Figure 129. A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs (disjoin vertices). The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²).

(3) Block perimeter

As blocks become larger, their perimeter also tends to become larger. For all the blocks analyzed, the block perimeter ranges from 400 to 13000 m. As shown in Figure 130B, when analyzed in groups, the maximum block perimeter observed in each group increases as *T* increases. The mean block perimeter per design also increases as T increases. More specifically, the mean block perimeter for a design generated by applying the operation of disjoining vertices *T* times on the initial 9 × 9 square-grid design is [2 × 100(144 – *T*) – 800 × 4] / (64 – *T*) m, or (25600 – 200*T*) / (64 – *T*) m.

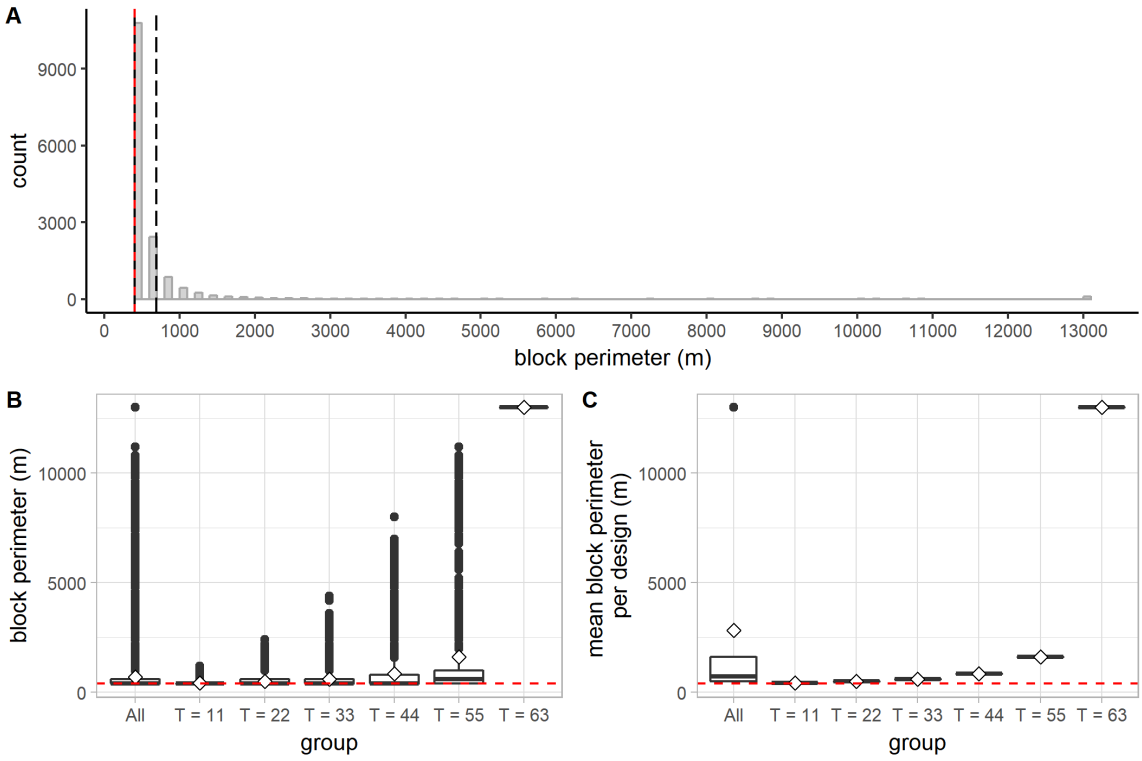

Figure 130. Distribution of the block perimeter (disjoin vertices). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

As the operation is applied more frequently, the block perimeter in a design also becomes more varied. As shown in Figure 131, the standard deviation of the block perimeter per design tends to increase as the operation is applied more frequently.

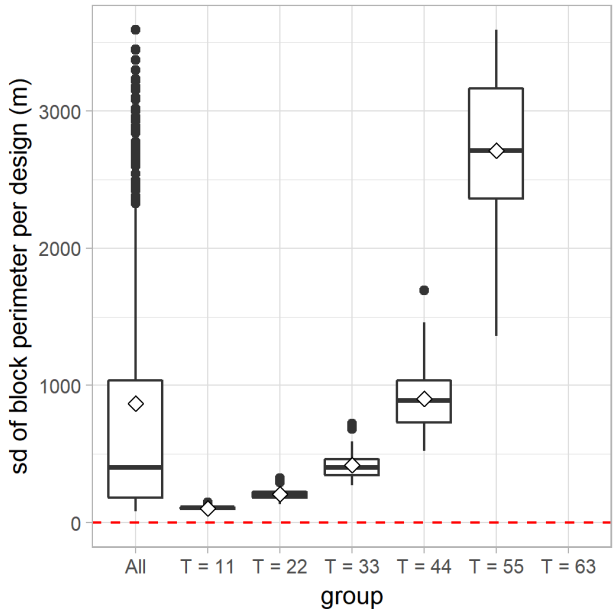

Figure 131. A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs (disjoin vertices). The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m).

(4) Standardized block area-perimeter ratio (SAPR)

For all the blocks analyzed, the standardized block area-perimeter ratio (SAPR) ranges from 0.25 to 1.00. As shown in Figure 132B, the more frequently the operation is applied, the more likely it is to generate extremely oddly shaped blocks. When analyzed in groups, as *T* increases, the mean SAPR per design tends to decrease as well.

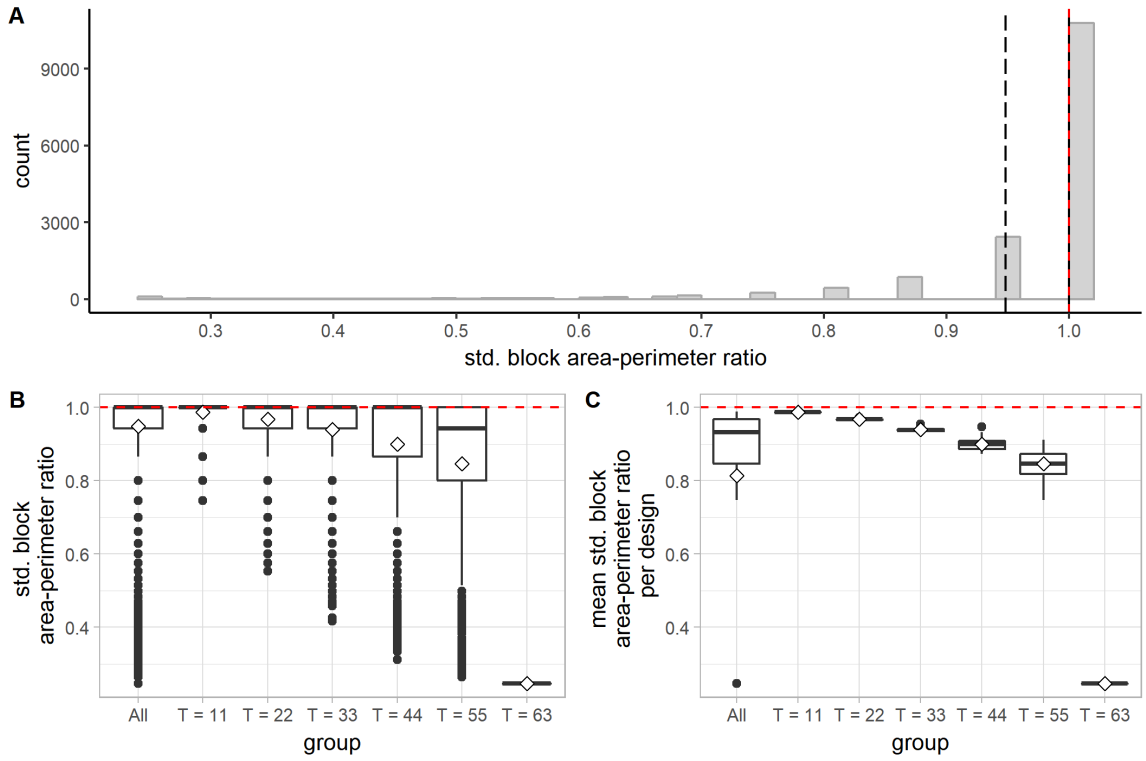

Figure 132. Distribution of the standardized block area-perimeter ratio (SAPR) (disjoin vertices). A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and the proportion of distinct DDL20d values per design

The total number of distinct DDL20d values per design ranges from 16 to 40. As shown in Figure 133B, the total number of distinct DDL20d values per design tends to increase as the operation is applied more frequently. To counteract the effect caused by the decreasing total number of segments, we analyzed the proportion of the distinct

DDL20d values per design. As shown in Figure 133C, when analyzed in groups, there is a clear upward trend in the proportion of distinct DDL20d values per design as *T* increases.

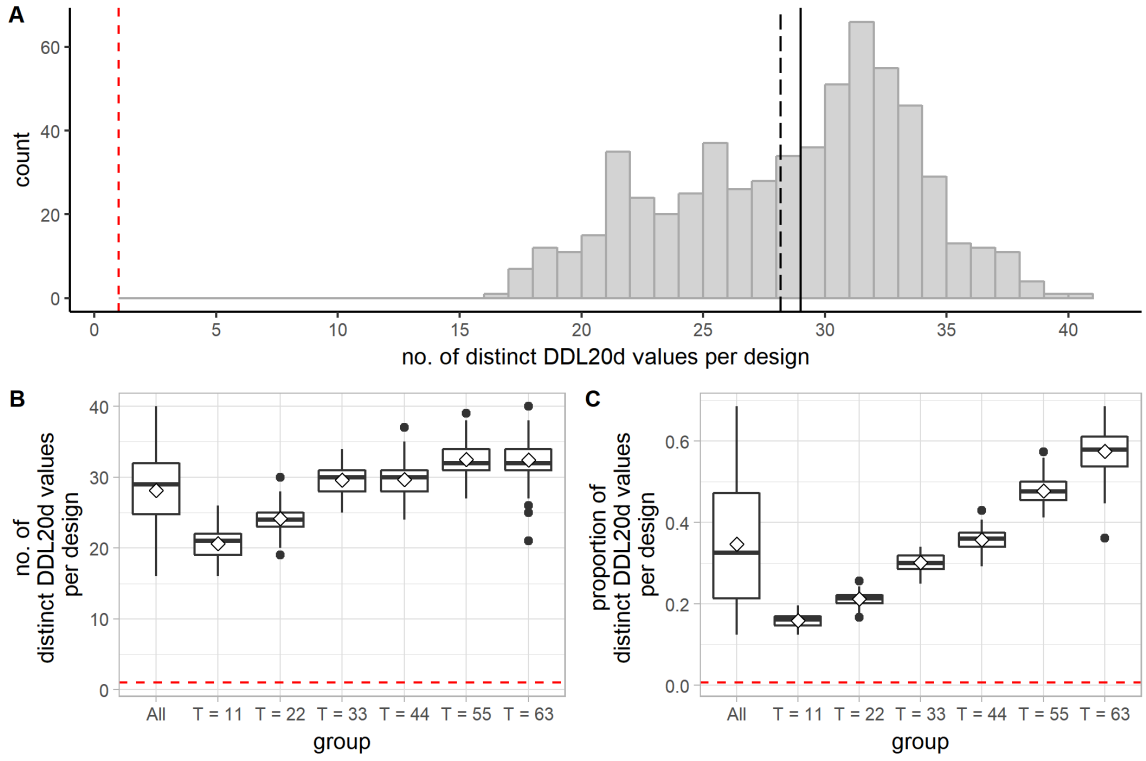

Figure 133. Distribution of the total number and the proportion of distinct DDL20d values per design (disjoin vertices). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design ($\approx$ 0.007).

(2) Standard deviation of DDL20d values per design

For all the designs analyzed, the standard deviation of DDL20d per design ranges from 0.22 to 2.00. In more than half cases, the standard deviation of DDL20d per design is greater than 0.5 (Figure 134A). As shown in Figure 134B, when analyzed in groups, as *T* increases, the standard deviation of DDL20d per design tends to increase (and increase at a faster rate as *T* increases).

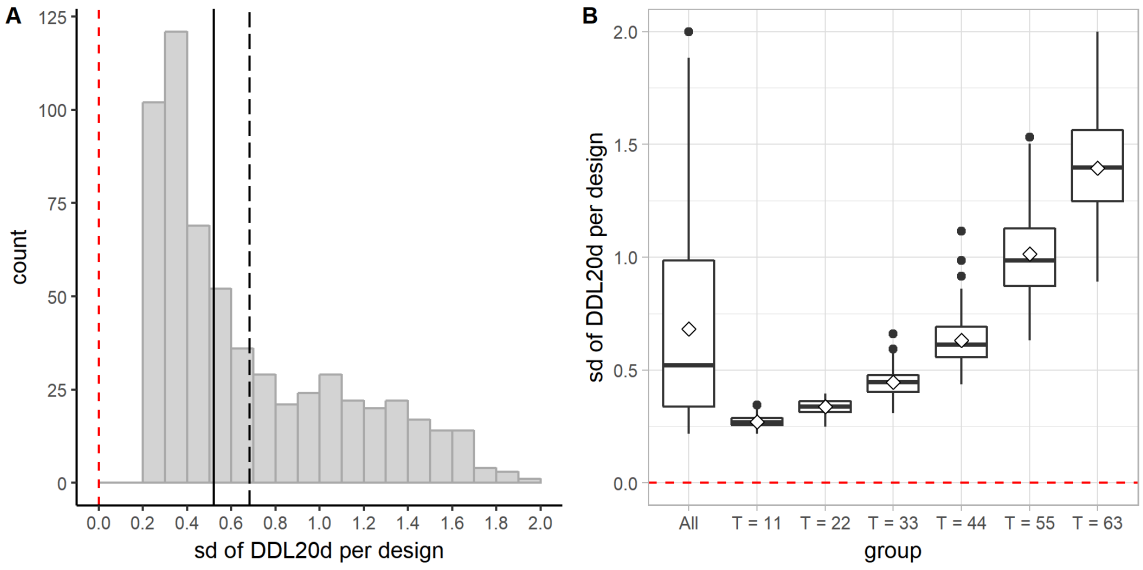

Figure 134. Distribution of the standard deviation of DDL20d per design (disjoin vertices). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

7.4 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: The operation maintains the total number of vertices per design. Every single application of the operation decreases the total number of edges by one and decreases the total number of cells by one as well. The operation decreases the vertex degree by removing the incident edges of the affected vertices. As a result, the operation also creates cul-de-sacs, but will never make the street graph disconnected.

The impact of the operation on the density of a street network: The operation decreases the total street length and the total number of blocks per design. It also tends to decrease the total number of intersections. By reducing the number of intersections per design, the operation also tends to increase the mean distance between intersections per design.

The impact of the operation on the directional reach/distance: By removing edges, this operation disrupts the continuity of the original street network. In doing so, it can create streets that are extremely segregated in terms of the directional distance. The operation tends to increase the mean DDL20d per design dramatically. The linear extensions of streets are disrupted by this operation, as a result, the operation tends to decrease the mean linear reach per design. It also tends to decrease the mean 2-dc reach per design.

The impact of the operation on the regularity of a street network: The operation tends to increase the fragmentality per design, because the average continuity line comprises a fewer number of segments. Frequent application of this operation can lead to extremely big blocks and blocks with an extremely large perimeter. The operation also

tends to make the block area and perimeter more varied in a superblock design. Frequent application of this operation also tends to result in blocks that have extremely low SAPR, or very oddly shaped blocks.

The impact of the operation on the diversity of syntactic conditions: This operation tends to increase the diversity of the syntactic conditions in a superblock, especially viewed from the perspective of the proportion of distinct DDL20d values per design. It tends to increase the standard deviation of DDL20d per design, largely because of the existence of the extremely segregated streets.

CHAPTER 8
LINKING VERTEX TO EDGE AND LINKING EDGE TO EDGE: SUBDIVIDING URBAN BLOCKS

Part A: Link Vertex to Edge

8.1 Operation: Link Vertex to Edge

8.1.1 Definition of operation

The operation of linking vertex to edge adds an edge to connect an existing vertex and an existing edge that belong to the same cell (Figure 135). In doing so, we create a new vertex on the edge to be linked to and split the affected cell into two.

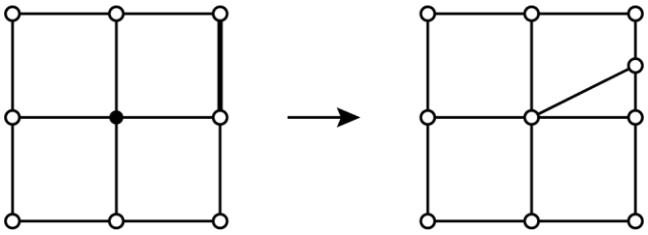

Figure 135. Link a vertex to an edge.

8.1.2 Parameters

To precisely control the operation of linking vertex to edge, we need to specify a point on the edge to be linked to. For example, as shown in Figure 136, to link the vertex *A* to the edge {*B*, *C*}, we need to know where to place the new vertex *D* on the edge {*B*, *C*} so that we can link the vertex *A* and the edge {*B*, *C*}. Instead of specifying the distance between *D* and *B*, or *d*(*B*, *D*), we can specify the ratio between *d*(*B*, *D*) and *d*(*B*,

C). To avoid ambiguity, we can use the half-edge *h* whose origin vertex is *B* to indicate both the edge to be linked to as well as the reference point for the distances involved.

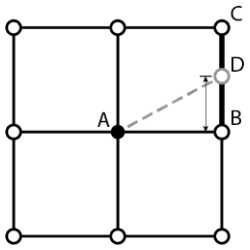

Figure 136. Parametric control of linking vertex to edge.

8.2 Generative Algorithms

8.2.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock design, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by *X*;
- the number of streets running horizontally in the initial regular grid, denoted by *Y*;
- the vertex to be linked, as indicated by a half-edge h_1;
- the edge to be linked, as indicated by a half-edge h_2;
- the minimum ratio between the distance from the newly created vertex on the edge $h_2.e$ to the vertex $h_2.twin\_h.v$ and the length of the edge $h_2.e$, denoted by r_{min};

- the maximum ratio between the distance from the newly created vertex on the edge $h_2.e$ to the vertex $h_2.twin\_h.v$ and the length of the edge $h_2.e$, denoted by r_{max};
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled by the following parameters: l_x, l_y, X, Y, h_1, h_2, r_{min}, r_{max}, T.

8.2.2 General description

We start from a street graph that represents a regular grid. Then we randomly pick a cell c in the street graph. We further randomly pick two half-edges, h_1 and h_2, both of which belong to the cell c, and attempt to link the vertex $h_1.v$ to a point on the edge $h_2.e$. We continue to do so until the operation has been successfully performed a specified number of times. The operation can only be successfully performed if *all* of the following conditions are met.

1. The area of the cell c is equal to the area of a block in the initial street graph—that is, $A(c) == l_x\, l_y / [(X - 1)(Y - 1)]$.
2. The length of the edge to be linked, $h_2.e$, is equal to the length of the block in the initial street graph—that is, $L(h_2.e) == max\{l_x / (X - 1), l_y / (Y - 1)\}$.
3. Linking the existing vertex $h_1.v$ to the newly created vertex on the edge $h_2.e$ meets all the conditions set for the operation of linking vertex to vertex (as described in Chapter 6).

The first two conditions are included to prevent the generation of blocks that are extremely small and distances between intersections that are extremely short. The above

conditions ensure that each of the original blocks in the initial square-grid design will be subdivided at most once and that each of the edges in the initial street graph will be linked at most once.

8.2.3 Pseudocode

Suppose that we have already generated the initial street graph G which represents a regular grid that is l_x units long, l_y units wide, with X number of streets running vertically and Y number of streets running horizontally. The algorithm developed to generate the superblock design with the operation of linking vertex to edge based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random* (a, b) refers to a procedure which returns a random floating-point number N such that $a \leqslant N \leqslant b$ for $a \leqslant b$, and *random_choice* (s) refers to a procedure which returns a random element from the set s.

ALGORITHM 8.1: Link Vertices to Edges

```
procedure link_vertices_to_edges (G, r_min, r_max, T)
counter ≔ 0
while counter < T
        c ≔ random_choice (C)
        if A(c) == l_x l_y / [(X − 1)(Y − 1)] then
                H_c ≔ a set comprising all half-edges that belong to c
                h_1 ≔ random_choice (H_c)
                H_c' ≔ H_c − {h_1, h_1.next_h}
                h_2 ≔ random_choice (H_c')
                if L(h_2.e) == max{l_x / (X − 1), l_y / (Y − 1)} then
                        r ≔ random (r_min, r_max)
                        vec_h_2 ≔ (h_2.v.x − h_2.twin_h.v.x, h_2.v.y − h_2.twin_h.v.y)
                        p ≔ (h_2.twin_h.v.x + r * vec_h_2.x, h_2.twin_h.v.y + r * vec_h_2.y)
                        perform the operation of linking h_1.v to p
                        counter ≔ counter + 1
```

Note that the restrictions we set for the operation of linking vertex to vertex also apply here when we attempt to link the vertex $h_1.v$ to the newly created point p on the edge $h_2.e$. If the operation cannot be performed, the point p will be removed, and the street graph will be restored to the state at the end of the previous attempt.

8.3 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the six groups of designs, we varied only one control parameter—namely T, the total number of operations to be performed—and kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9, r_{min} = 0.5, r_{max} = 0.5. By assigning the same value (0.5) to both r_{min} and r_{max}, we require that the vertex is always linked to the midpoint of the edge specified.

8.3.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation on the initial 9 × 9 square-grid design—that is, the values of T. We adopted the following strategy: First, we chose an arbitrarily high number a such that, based on our algorithm, it would be impossible to apply this operation more than a times on the initial design. Then we ran the algorithm by setting $T = a$ and applied the following terminating condition: If 20000 attempts have been made in a row and none was successful, then stop making any more attempts and record the total number of times that the operation has been successfully performed—let's denote it by b. It gave us a sense of the upper limit for the number of successful applications of the operation for a particular run.

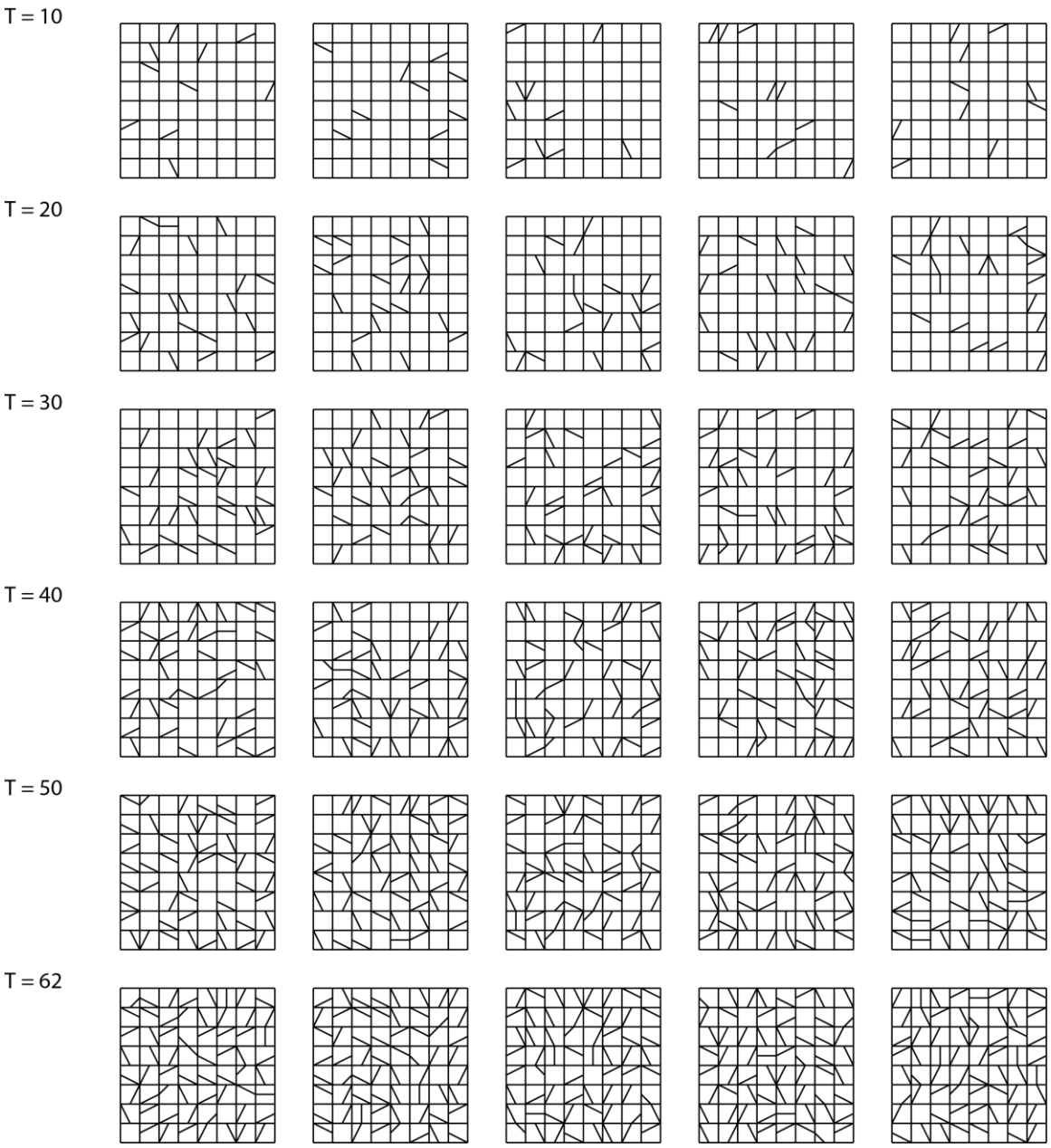

Figure 137. Examples of designs from each group (link vertex to edge).

Notice that the number *b* may vary for different runs, thus we made 100 trial runs and recorded the value of *b* after each trial. Then we chose the smallest recorded value as the maximum number of times to apply the operation in the generative process—let's denote it by T_{max}. The minimum number of times to apply the operation, T_{min}, was

subsequently determined by finding the nearest integer for T_{max} / 6. To generate the other four groups of designs, we simply set T to $2T_{min}$, $3T_{min}$, $4T_{min}$, and $5T_{min}$, respectively.

Examples of designs from each group are shown in Figure 137.

8.3.2 Data analysis

Six groups of designs were generated by applying the operation of linking vertex to edge 10, 20, 30, 40, 50, and 62 times, respectively, on the initial 9 × 9 square-grid design. Each group consists of 100 designs. The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of designs.

Elementary graph properties

(1) Number of vertices

By definition, each application of the operation creates a new vertex on the edge to be linked to. Therefore, as shown in Figure 138, the total number of vertices per design increases as the operation is applied more frequently. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, its number of vertices, $|V|$, equals $81 + T$.

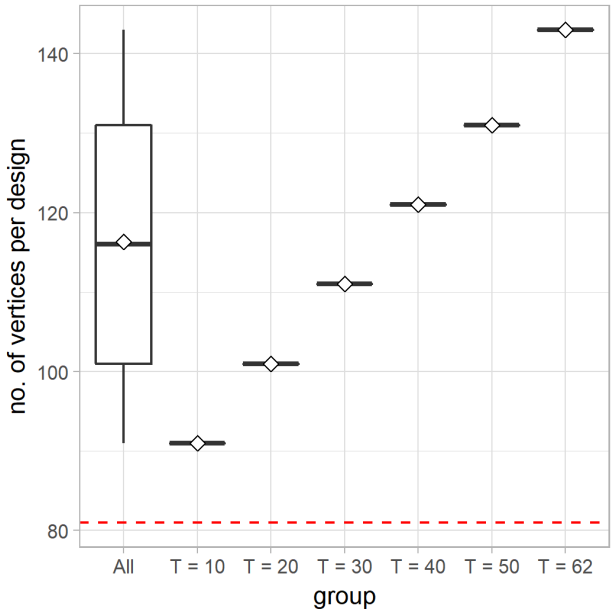

Figure 138. A boxplot (with mean diamonds) showing the distribution of the total number of vertices per design for all designs and for each group of designs (link vertex to edge). The red dashed line indicates the total number of vertices in the initial 9 × 9 square-grid design (81).

(2) Number of edges

Each time we apply the operation of linking vertex to edge, we create a new vertex on the edge to be linked to. And in doing so, we split the original edge into two edges, increasing the number of edges by one. As we also add an additional edge to link the specified vertex and edge, we increase the total number of edges by two each time we apply the operation. Therefore, for a design generated by applying the operation of linking vertex to edge T times on the initial 9 × 9 square-grid design, the total number of edges, $|E|$, equals $144 + 2T$. As shown in Figure 139, the total number of edges per design increases steadily as the operation is applied more frequently.

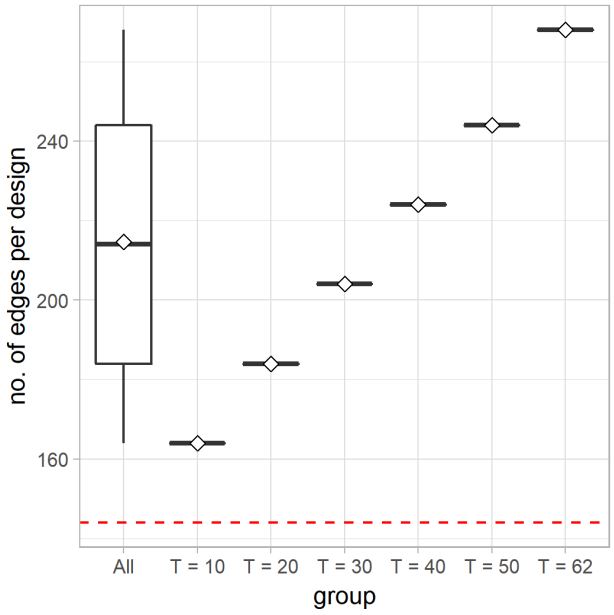

Figure 139. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (link vertex to edge). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

By definition, each time we apply the operation of linking vertex to edge, we also split an existing cell into two cells, thus increasing the total number of cells by one. As shown in Figure 140, the total number of edges per design increases steadily as the operation is applied more frequently. To be more specific, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the total number of cells, $|C|$, equals $64 + T$.

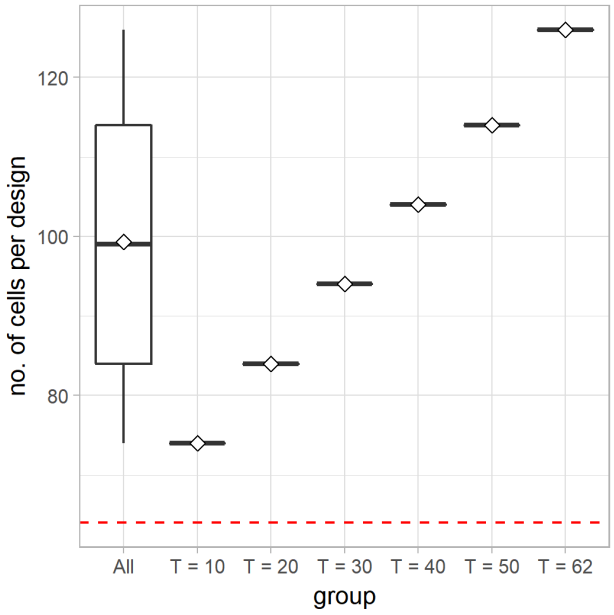

Figure 140. A boxplot (with mean diamonds) showing the distribution of the total number of cells per design for all designs and for each group of designs (link vertex to edge). The red dashed line indicates the total number of cells in the initial 9 × 9 square-grid design (64).

(4) Vertex degree

The operation of linking vertex to edge increases the degree of the vertex it links by one and contributes a new vertex of degree 3. For all the vertices analyzed, the vertex degree ranges from 2 to 8. As the operation is applied more frequently, vertices of degree 3 (or T-intersections) occur more frequently (notice the shift of median from 4 to 3 in Figure 141B). As shown in Figure 141C, the mean vertex degree per design increases steadily as the operation is applied more frequently. By the handshaking theorem, the sum of the degrees of the vertices of a graph is twice the number of its edges. Therefore, the mean vertex degree for a graph is $2|E| / |V|$. For a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the mean vertex degree is $2(144 + 2T) / (81 + T)$.

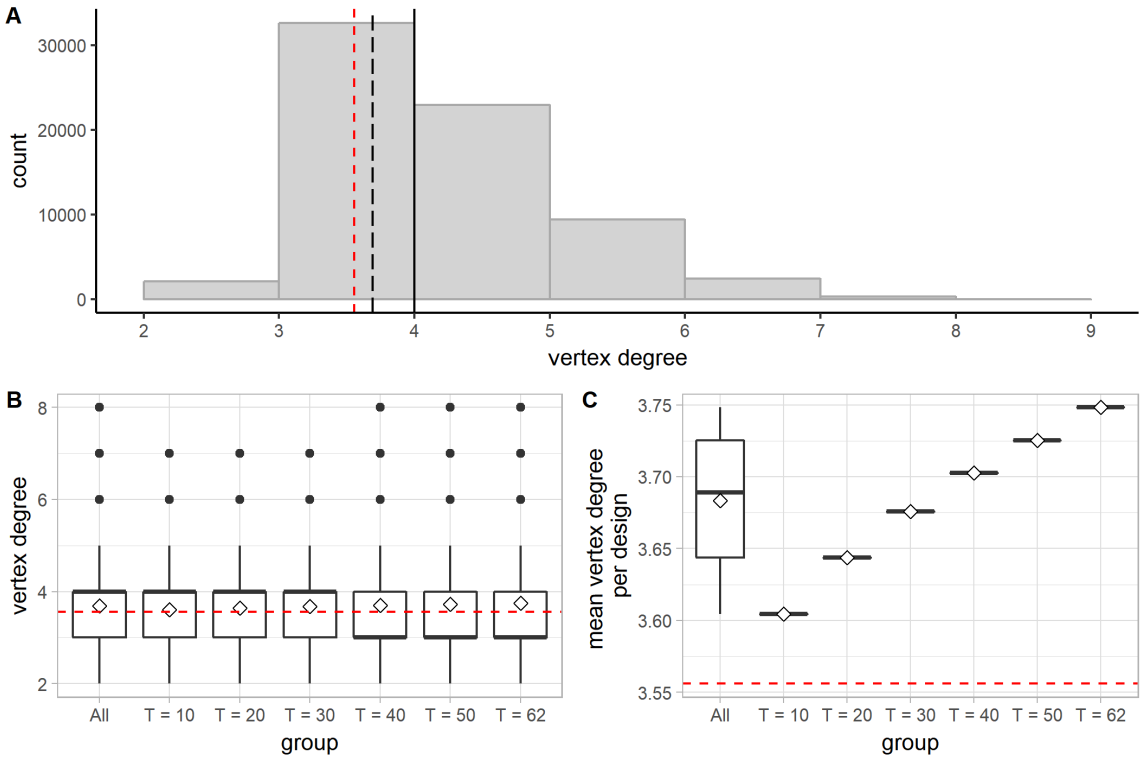

Figure 141. Distribution of the vertex degree (link vertex to edge). A: A histogram showing the distribution of the vertex degree for all vertices in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design ($\approx$ 3.556). B: A boxplot (with mean diamonds) showing the distribution of the vertex degree for all vertices and for vertices in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

By definition, the operation of linking vertex to edge adds additional edges and thus increases the total street length in a design. As shown in Figure 142, the total street length per design increases as the operation is applied more frequently.

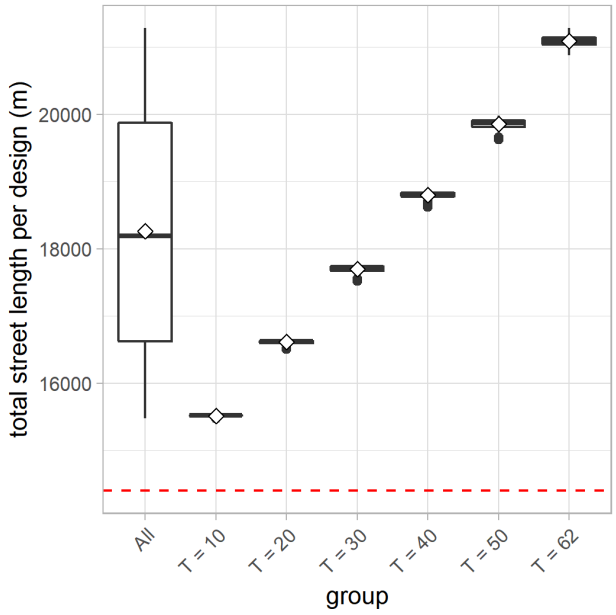

Figure 142. A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs (link vertex to edge). The red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m).

(2) Total number of blocks per design

The total number of blocks per design is the same as the total number of cells per design. It increases as the operation is applied more frequently.

(3) Total number of intersections per design

By definition, each application of operation creates a new vertex of degree 3 on the edge to be linked to—that is, a T-intersection. When one of the vertices located at the corners of the superblock is linked to a surrounding edge, an additional intersection will be created because the corner vertex itself—originally has degree 2—will be counted as an intersection when its degree is increased to 3 after being connected by a new edge. As

shown in Figure 143B, as the operation is applied more frequently, the total number of intersections per design tends to increase.

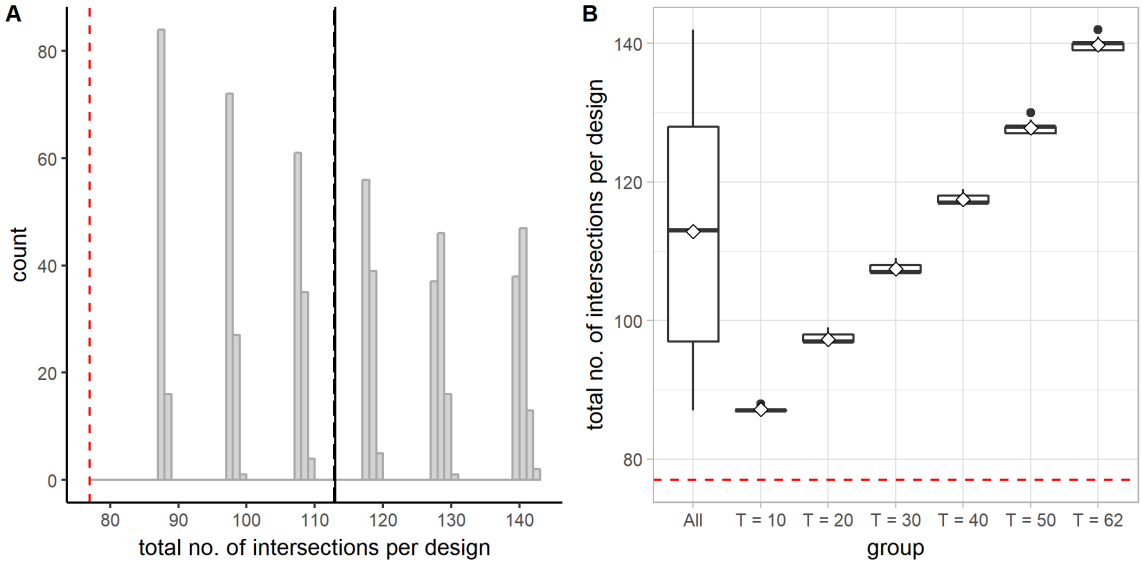

Figure 143. Distribution of the total number of intersections per design (link vertex to edge). A: A histogram showing the distribution of the total number of intersections per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77). B: A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs.

(4) Distance between intersections

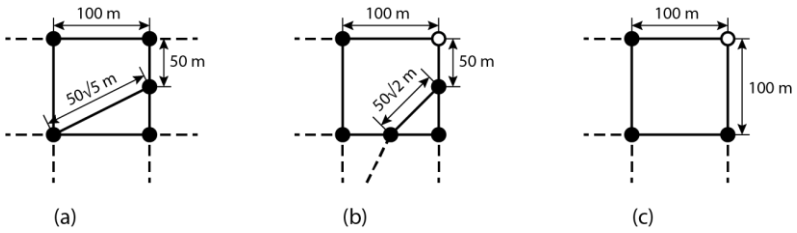

Figure 144. Illustration of possible distances between intersections (link vertex to edge). The vertices that have degree greater than or equal to 3 are colored in black.

As shown in Figure 144, based on our algorithm, the distance between intersections can only take one of six different values: 50 m, $50\sqrt{2}$ m, 100 m, $50\sqrt{5}$ m, 150 m, or 200 m. In the case shown in Figure 144a, after the operation of linking vertex to edge, the distance between intersections represented by the length of the oblique edge inside the square block is slightly longer than 100 m—that is, the typical distance between intersections in the original square-grid design. However, by splitting the edge on the right side of the square block in half, it significantly shortens the original distance between the two intersections on the corners of the square block. Therefore, on average it still reduces the distance between intersections.

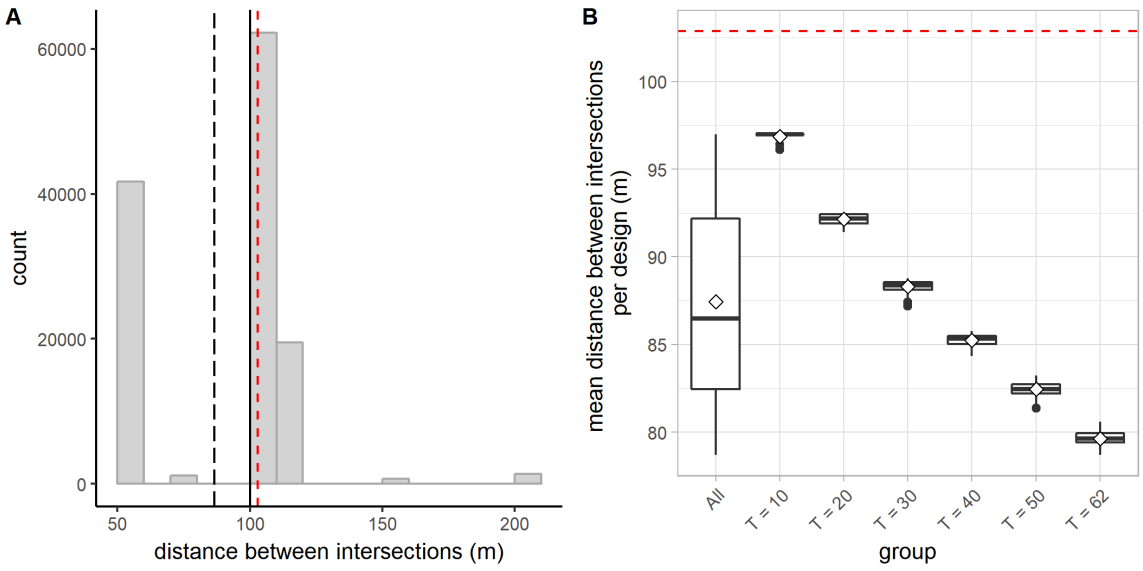

Figure 145. Distribution of the distance between intersections (link vertex to edge). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In both subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design ($\approx$ 102.86 m). B: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

As shown in Figure 145B, when analyzed in groups, as *T* increases, the mean distance between intersections per design tends to decrease.

Directional reach and directional distance

(1) DDL

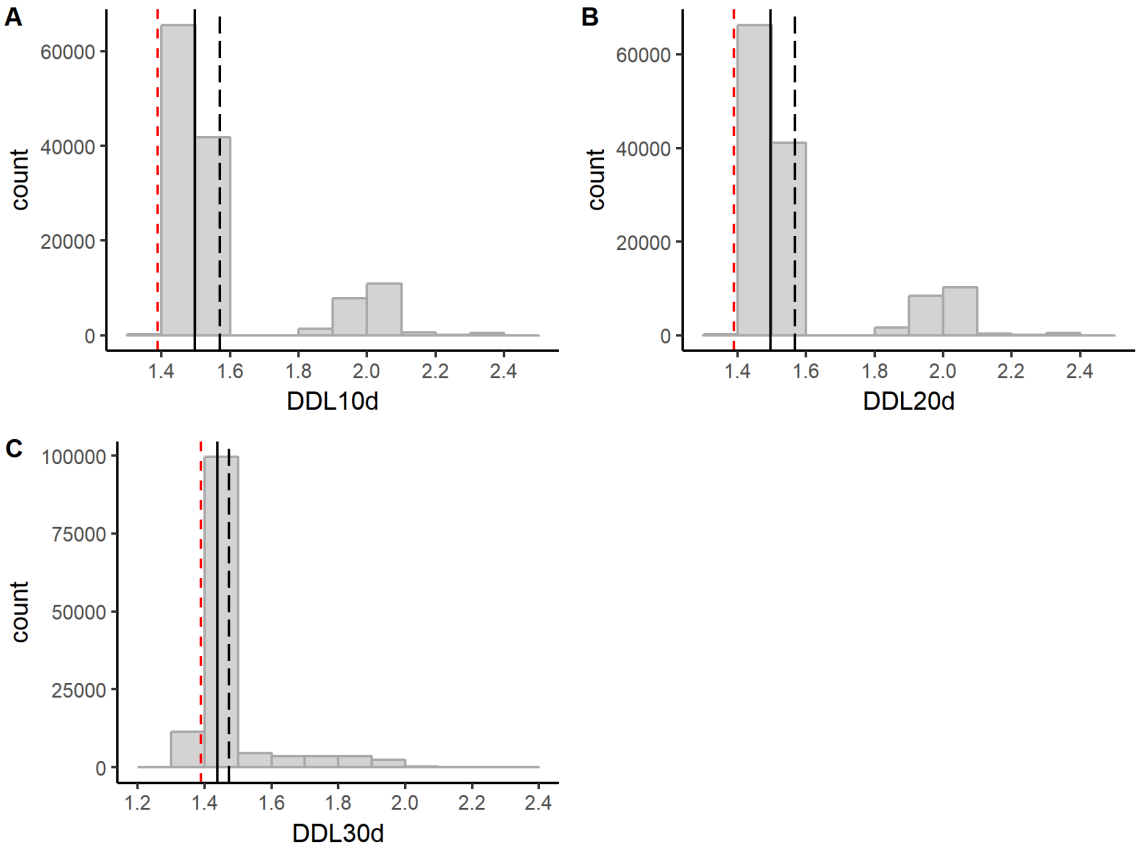

Figure 146. Distributions of DDL10d, DDL20d, and DDL30d (link vertex to edge). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

As shown in Figure 146, the distribution of the DDL values does not change much when the threshold angle is increased from 10° to 20°. There is a notable

difference, however, when the threshold angle is increased to 30°. In the following analysis, unless otherwise specified, the directional distance is always evaluated by setting the threshold angle to 20°. For all the segments analyzed, DDL20d ranges from 1.40 to 2.43. As shown in Figure 146B, the segment typically assumes a DDL20d value that is between 1.4 and 1.6.

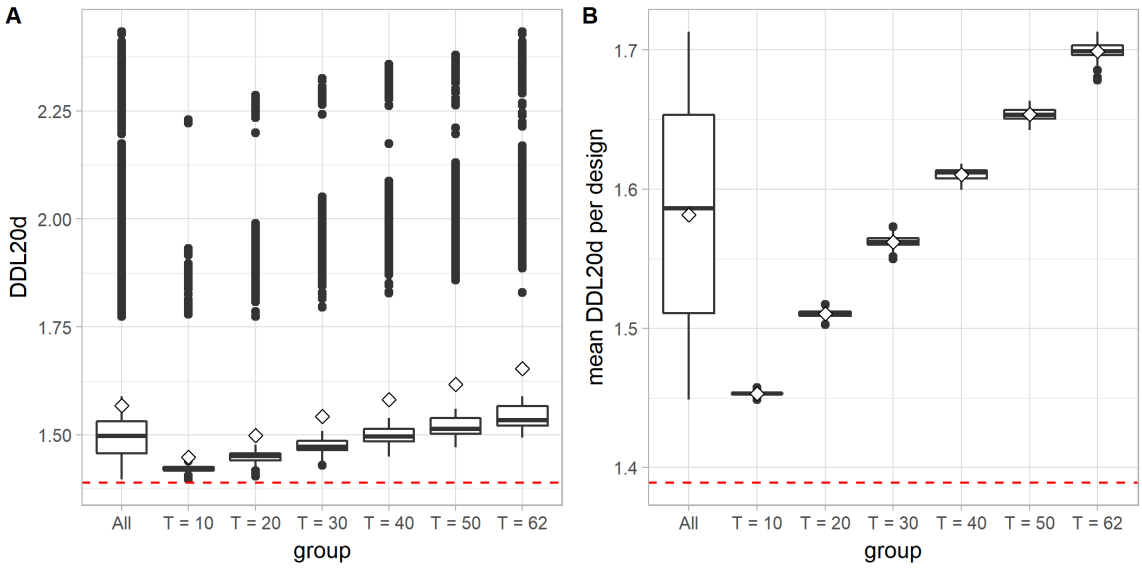

Figure 147. Distribution of DDL20d (link vertex to edge). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

The operation of linking vertex to edge tends to create edges that have a short linear reach. These edges are often relatively segregated from each other because, unless the edges are incident, they are at least two turns away from each other. By contrast, in most cases, it never takes more than two turns to get anywhere in the street graph if we start from a street existed in the initial grid. Therefore, based on our algorithm and the

setup of the initial street graph, the operation of linking vertex to edge tends to increase the mean directional distance of the street graph. As shown in Figure 147B, when analyzed in groups, the mean DDL20d tends to increase as *T* increases.

(2) Linear reach (dr0dc20d)

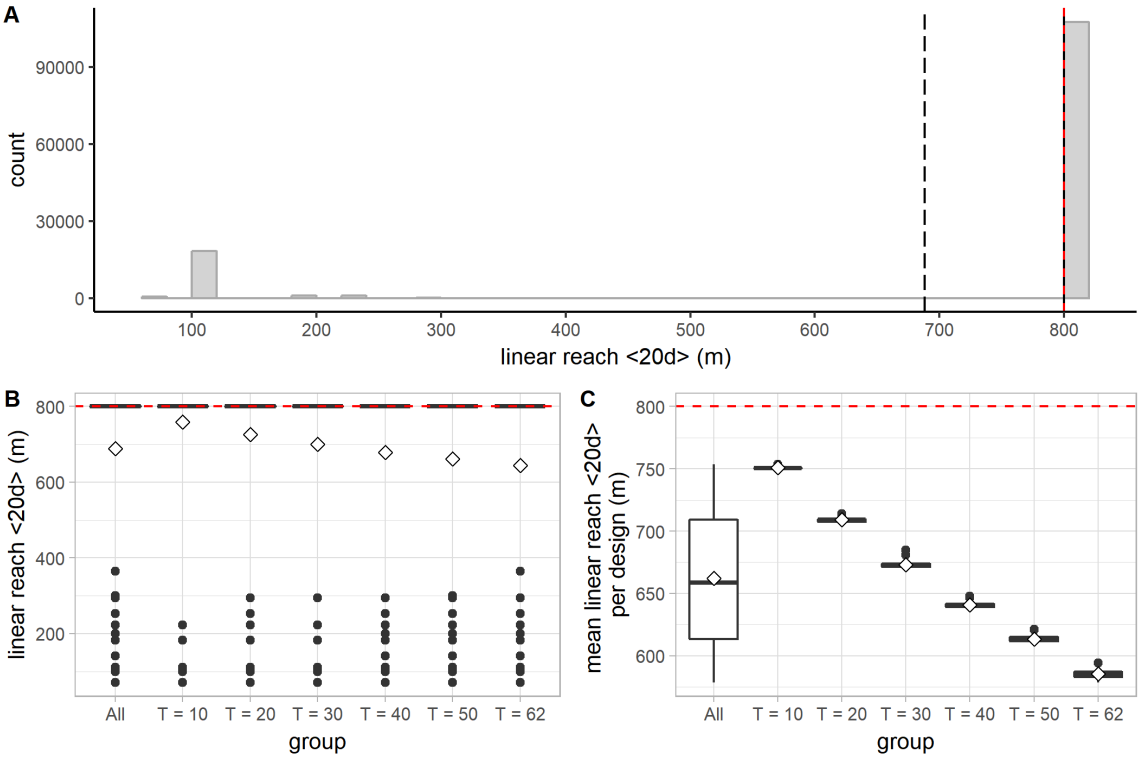

Figure 148. Distribution of the linear reach (link vertex to edge). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

As shown in Figure 148, for most of the segments analyzed, their linear reach is 800 m. This is hardly surprising because most segments are the components of the

horizontally- and vertically-running streets existed in the initial square-grid design and their linear reach is always 800 m. The other linear reach values reflect the lengths of the edges inside the square blocks and the sums of the lengths of the edges (if they are incident and aligned linearly with each other). As shown in Figure 148B, as the operation is applied more frequently, more segments with a short linear reach are generated. When analyzed in groups, as *T* increases, the mean linear reach per design tends to drop as well (Figure 148C).

(3) 2-dc reach (dr2dc20d)

As shown in Figure 149, most segments assume a 2-dc reach value greater than 14400 m—that is, the total street length for the initial square-grid design. It is not surprising because for most segments—even those that are not located on the initial grid—they can reach the entire street network in the initial square-grid design within two direction changes. Only when the edge inside the square block is placed parallel to the side of the block can its 2-dc reach fall short of the total street length of the initial design. As shown in Figure 149B, for most segments in the design, their 2-dc reach tends to increase as the operation is applied more frequently—partially because of the increased total street length in the design. As a result, when analyzed in groups, as *T* increases, the mean 2-dc reach per design tends to increase as well (Figure 149C).

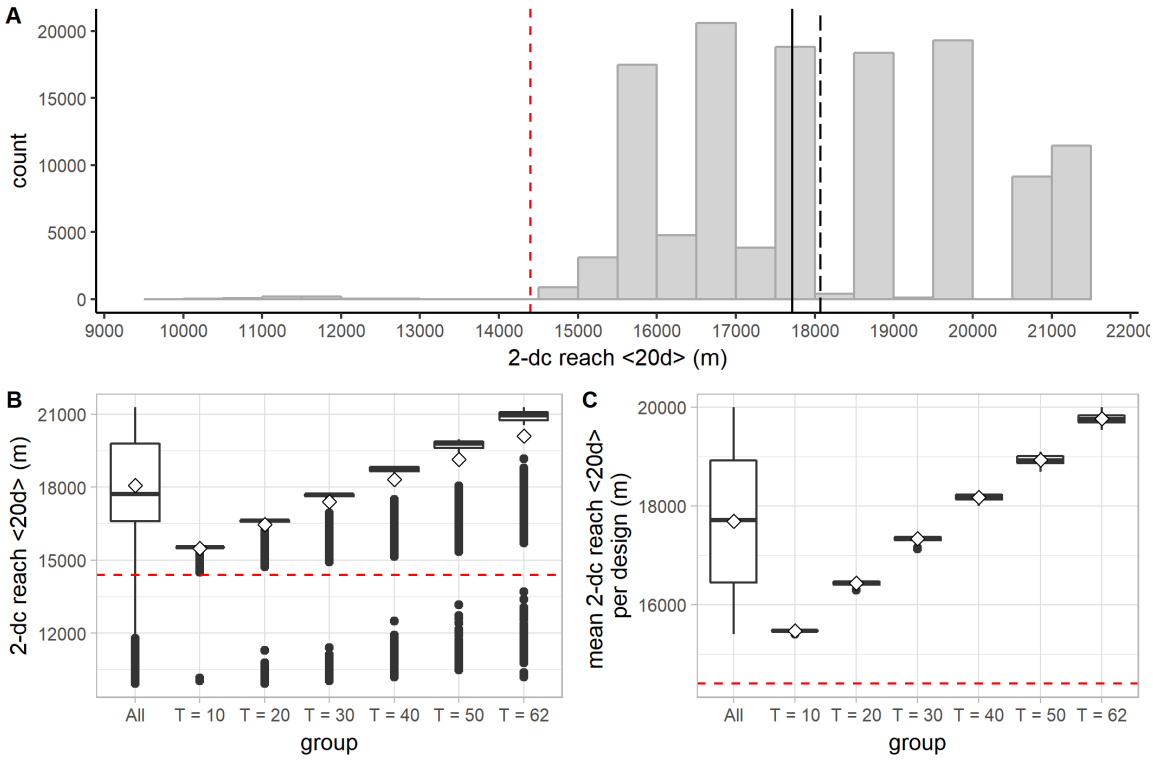

Figure 149. Distribution of the 2-dc reach (link vertex to edge). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

The fragmentality per design ranges from 0.16 to 0.30. Since the added edges inside the square blocks are rarely aligned linearly with each other, the design tends to be more fragmented as the operation is applied more frequently. As shown in Figure 150B,

when analyzed in groups, as *T* increases, the fragmentality per design tends to increase as well.

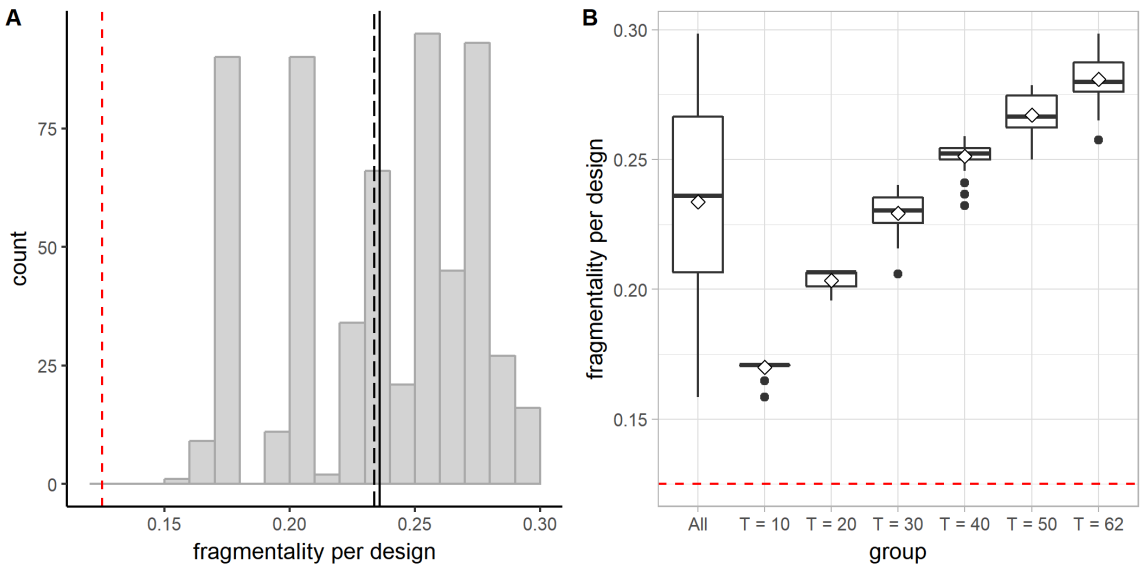

Figure 150. Distribution of the fragmentality per design (link vertex to edge). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

Based on our algorithm, the block area can only take one of the following values: 10000 m², 2500 m², 7500 m², 1250 m², 8750 m², and 5000 m², corresponding to the areas of the various kinds of blocks resulted from the subdivision of the square blocks. Obviously, as the operation is applied more frequently, more blocks will be subdivided into smaller blocks, thus reducing the mean block area per design. As shown in Figure 151C, when analyzed in groups, as *T* increases, the mean block area per design steadily decreases. More specifically, for a design generated by applying the operation of linking

vertex to edge *T* times on the initial 9 × 9 square-grid design, the mean block area is 800 × 800 / (64 + *T*) m², or 640000 / (64 + *T*) m².

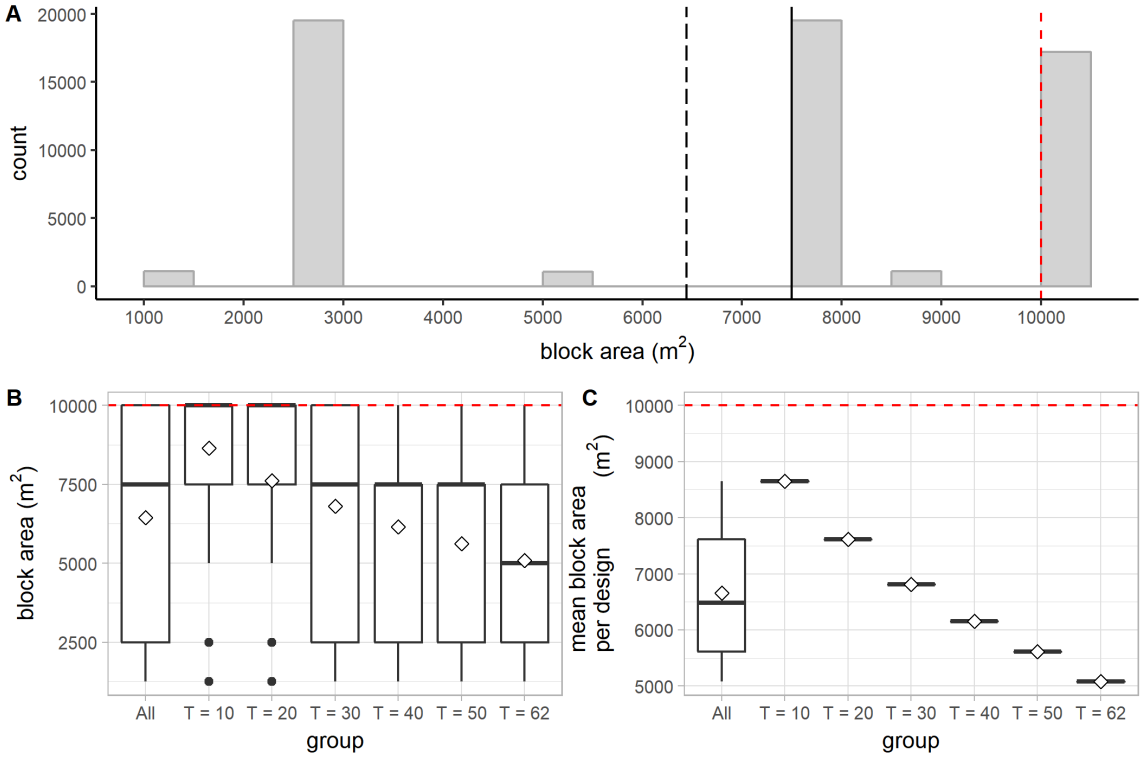

Figure 151. Distribution of the block area (link vertex to edge). A: A histogram showing the distribution of the area for all the blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²). B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs. C: A boxplot showing the distribution of the mean block area per design for all designs and for each group of designs.

As shown in Figure 152A, the standard deviation of the block area per design tends to grow at first and drop after a certain point. One reason for the downward trend is that the mean block area gets much lower and that the difference between the block sizes, in terms of magnitude, becomes smaller. To analyze the relative variation of the block area, we computed the coefficient of variation (CV) for the block area per design and it

shows that the CV for block area per design tends to increase as the operation is applied more frequently (Figure 152B).

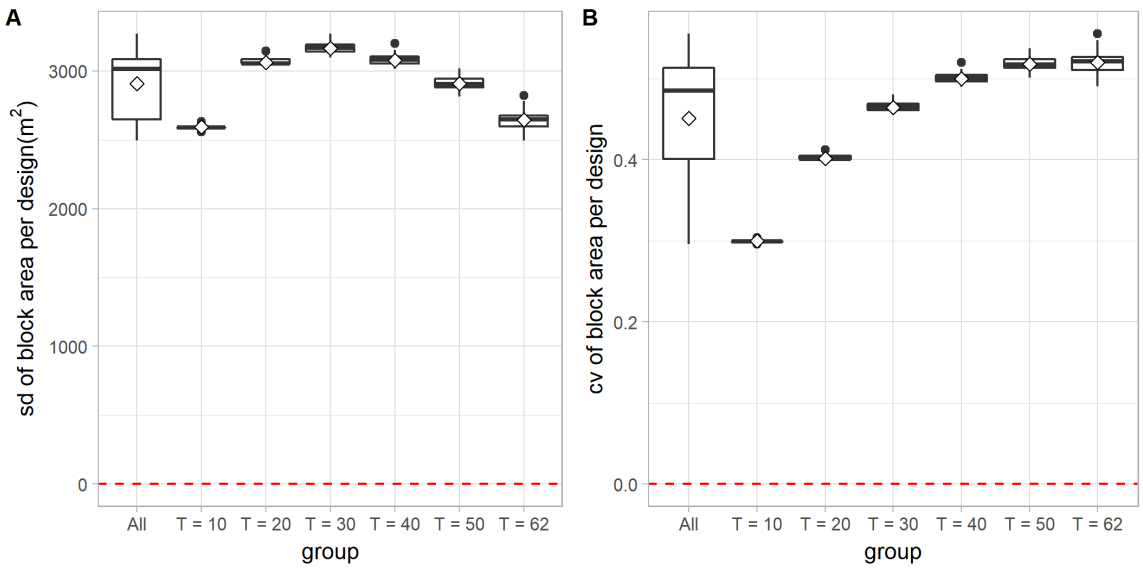

Figure 152. Variation of the block area per design (link vertex to edge). A: A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs. The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²). B: A boxplot showing the distribution of the coefficient of variation (CV) of the block area per design for all designs and for each group of designs. The red dashed line indicates the CV of the block area per design for the initial 9 × 9 square-grid design (0.0).

(3) Block perimeter

Based on our algorithm, the block perimeter can only take one of the following six values: 400 m, 300 m, $150 + 50\sqrt{5}$ m, $100 + 50\sqrt{2}$ m, $250 + 50\sqrt{5}$ m, and $300 + 50\sqrt{2}$ m, corresponding to the perimeters of the various kinds of blocks resulted from the subdivision of the square blocks. As the operation is applied more frequently, more blocks are subdivided into smaller blocks and their perimeter reduced. As shown in

Figure 153C, when analyzed in groups, the mean block perimeter per design decreases as *T* increases.

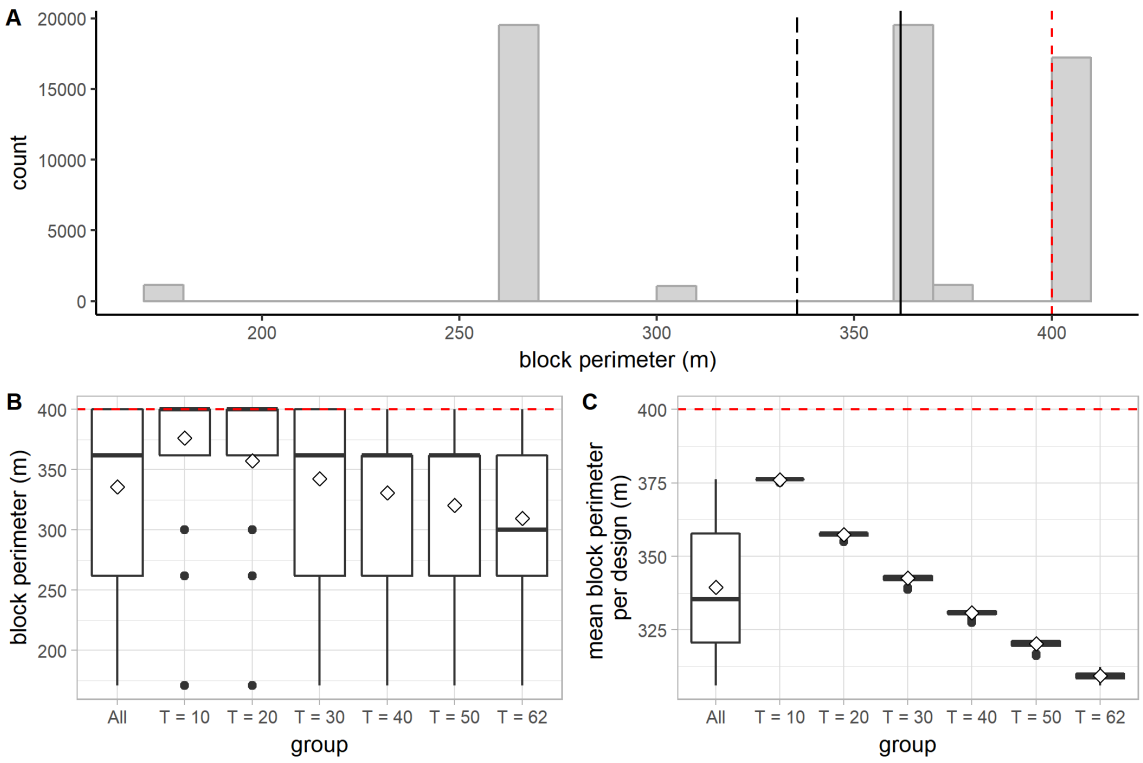

Figure 153. Distribution of the block perimeter (link vertex to edge). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

As shown in Figure 154A, the standard deviation of the block perimeter per design tends to increase at first but then tends to decrease after a certain point. One reason for the downward trend could be that the mean block perimeter gets smaller and so does the magnitude of the difference between the block perimeters. To analyze the relative variation of the block perimeter, we computed the coefficient of variation (CV)

for the block perimeter per design and it shows that the CV of the block perimeter per design clearly tends to increase at the beginning. However, the upward trend disappears after a certain point (Figure 154B).

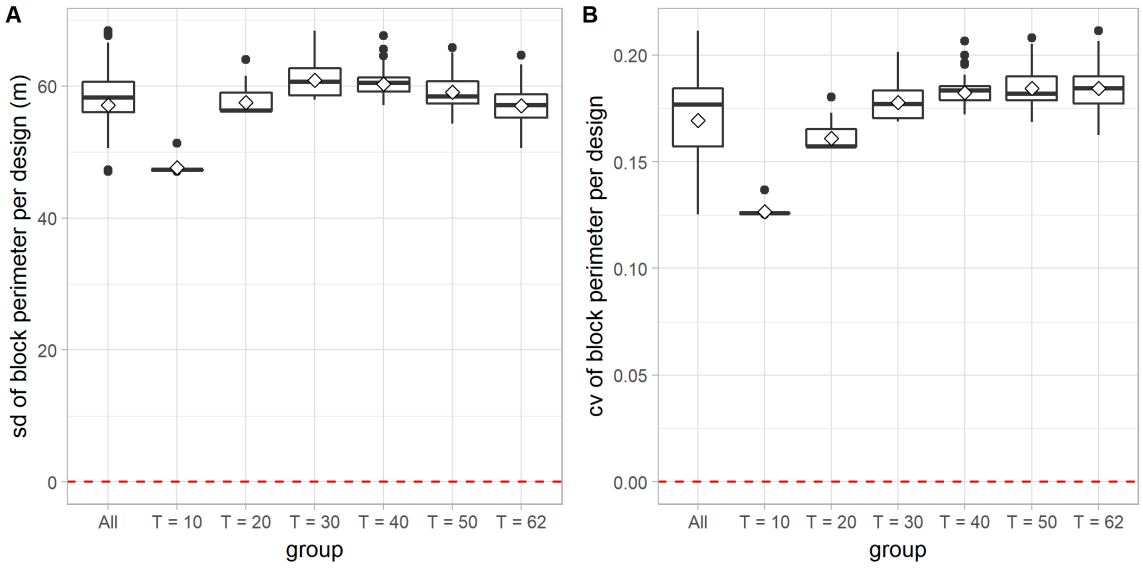

Figure 154. Variation of the block perimeter per design (link vertex to edge). A: A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs. The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m). B: A boxplot showing the distribution of the coefficient of variation (CV) of the block perimeter per design for all designs and for each group of designs. The red dashed line indicates the CV of the block perimeter per design for the initial 9 × 9 square-grid design (0.0).

(4) Standardized block area-perimeter ratio (SAPR)

The standardized block area-perimeter ratio (SAPR) can only take six different values, corresponding to the six different block shapes that can possibly be generated based on our algorithm. For all the blocks analyzed, the SAPR ranges from 0.76 to 1.01. More than 65% of all blocks have a SAPR greater than 0.9. As shown in Figure 155C, when analyzed in groups, as *T* increases, the mean SAPR per design decreases.

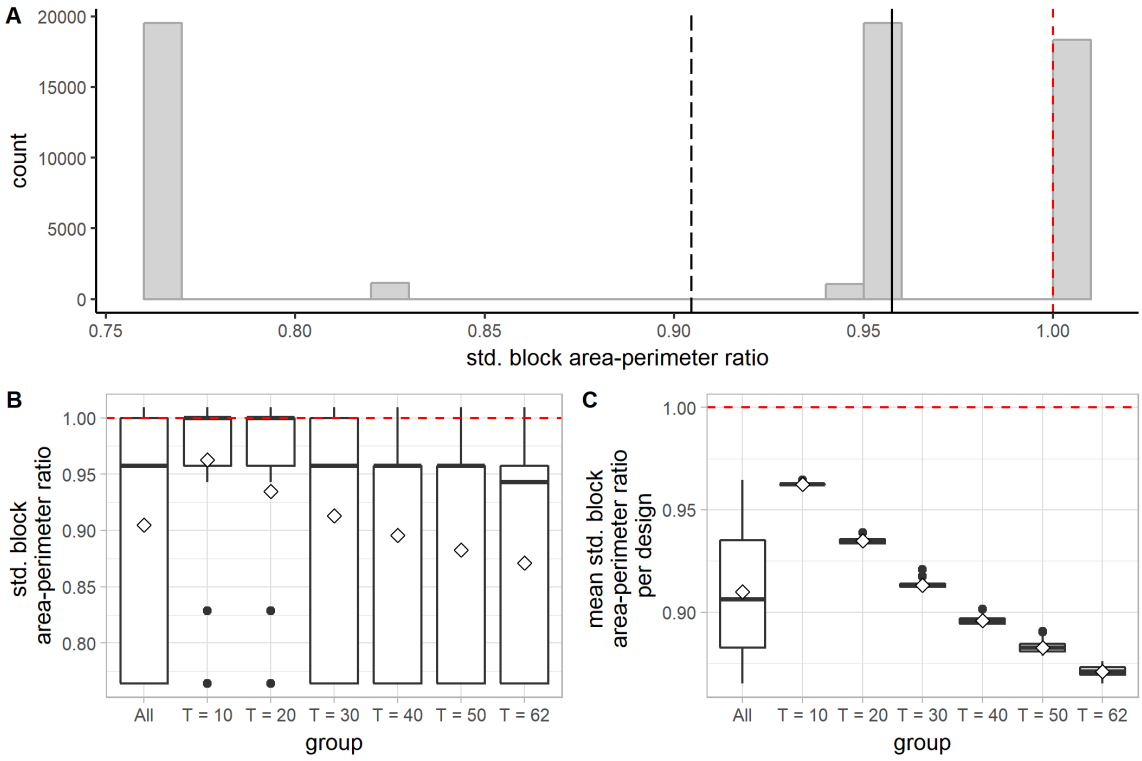

Figure 155. Distribution of the standardized block area-perimeter ratio (SAPR) (link vertex to edge). A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and proportion of distinct DDL20d values per design

The total number of distinct DDL20d values per design ranges from 7 to 73. The proportion of distinct DDL20d values per designs ranges from 0.04 to 0.27—a relatively small range. As shown in Figure 156 B and C, both the total number of distinct DDL20 values and the proportion of distinct DDL20d values per design tend to increase as the

operation is applied more frequently. However, there is generally a lack of diversity in syntactic conditions considering the low proportion of distinct DDL20d values.

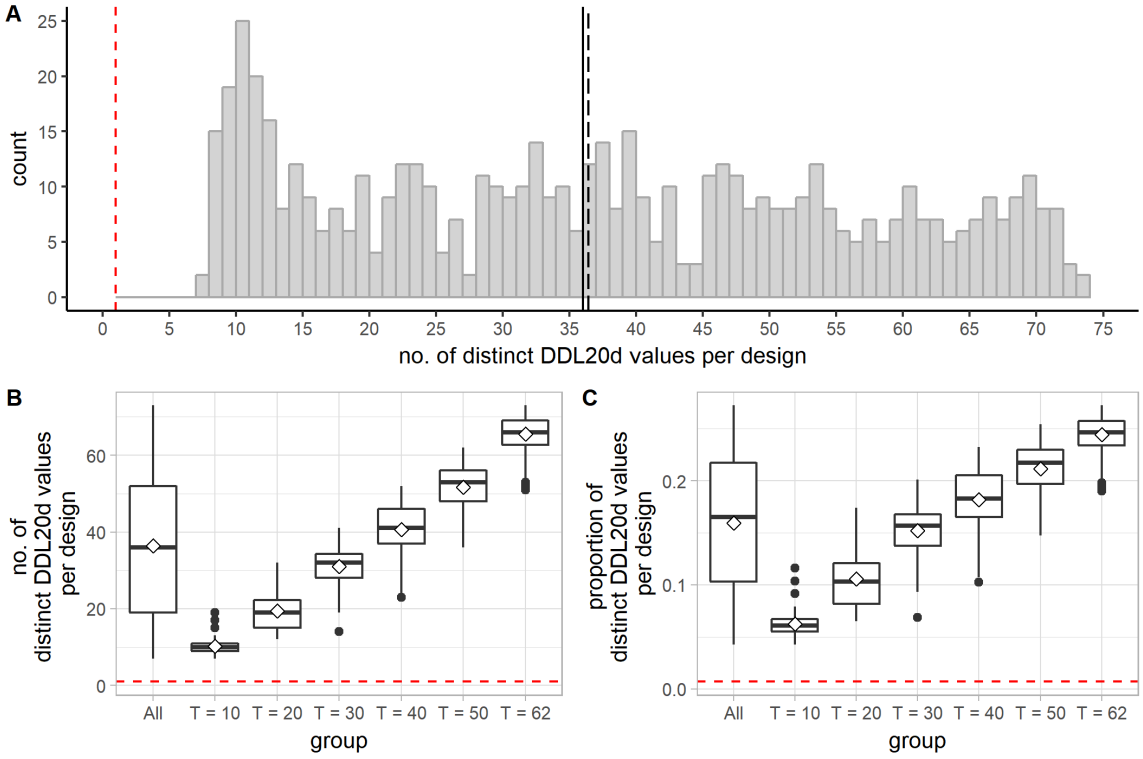

Figure 156. Distribution of the total number and the proportion of distinct DDL20d values per design (link vertex to edge). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d values per design

For all the designs analyzed, the standard deviation of DDL20d per design ranges from 0.11 to 0.26. As shown in Figure 157B, when analyzed in groups, as *T* increases, the standard deviation of DDL20d per design tends to increase.

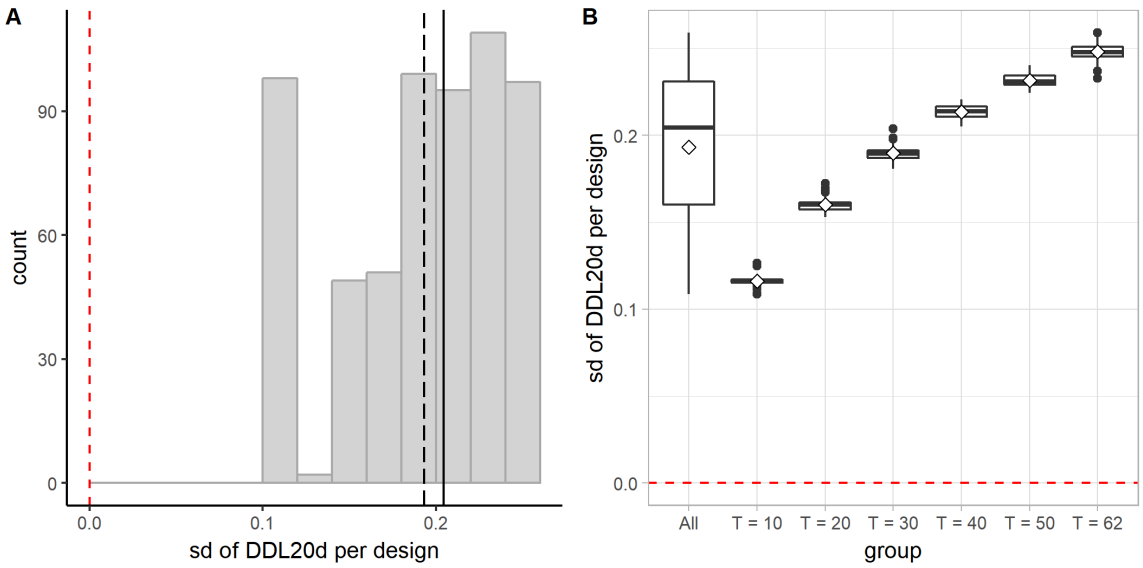

Figure 157. Distribution of the standard deviation of DDL20d per design (link vertex to edge). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

8.4 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: Each application of this operation increases the total number of vertices by one, the total number of edges by two, and the total number of cells by one. The operation increases the degree of existing vertices and creates new vertices of degree 3. Overall it tends to increase the mean vertex degree per design (although the mean vertex degree would never exceed 4, thus would not create a huge difference anyway).

The impact of the operation on the density of a street network: The operation increases the total street length per design, the total number of blocks per design, as well as the total number of intersections per design (although often in the form of T-intersections). Since most of the time it generates segments shorter than the segments in the initial square-grid design, the operation tends to reduce the mean distance between intersections per design.

The impact of the operation on the directional reach/distance: The operation tends to increase the mean DDL20d per design because the additional segments placed inside the original square blocks are relatively secluded from the rest of the system compared to the horizontal and vertical segments which comprise the initial grid. The operation tends to decrease the mean linear reach per design while increasing the mean 2-dc reach per design. The former trend is attributed to the additional segments placed inside the original

square blocks which usually have a very short linear reach, while the latter is partially attributed to the increasing total street length (i.e., a denser street network).

The impact of the operation on the regularity of a street network: This operation tends to increase the fragmentality per design. As the operation is applied more frequently, not only do the mean block area and the mean block perimeter per design tend to get smaller, the blocks in a design also tend to get more varied in terms of the size and perimeter, especially before the operation has affected more than half of the original blocks. The operation also tends to decrease the standardized block area-perimeter ratio per design, making blocks with shapes less compact than a perfect square.

The impact of the operation on the diversity of syntactic conditions: This operation tends to increase the diversity of the syntactic condition in a superblock design. However, in general, there is a lack of diversity in syntactic condition, as demonstrated by the low proportion of distinct DDL20d values per design. Nevertheless, the operation tends to make the syntactic conditions in a design more varied, as evidenced by the increasing standard deviation of DDL20d values per design.

Part B: Link Edge to Edge

8.5 Operation: Link Edge to Edge

8.5.1 Definition of operation

The operation of linking edge to edge connects two edges that belong to a common cell with a new edge (Figure 158). In doing so, it creates two new vertices, one on each edge it links, and splits the affected cell into two.

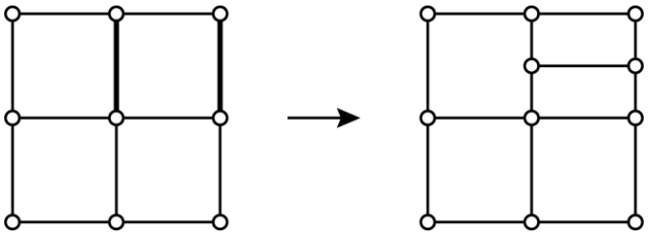

Figure 158. Link an edge to an edge.

8.5.2 Parameters

To precisely control the operation of linking edge to edge, we need to specify the position of the endpoints of the new edge we added to connect the existing edges and split the cell. For example, as shown in Figure 159, to link the edge {*A*, *B*} and the edge {*C*, *D*}, we need to specify the position of the vertices *E* and *F*—namely, the endpoints of the new edge {*E*, *F*} that we are adding. A position along an edge can be specified by the distance from one endpoint of the edge, or the ratio between the distance from one endpoint of the edge and the entire length of the edge. For instance, we can specify the position of the vertex *E* in Figure 159 by the ratio between *d*(*A*, *E*) and *d*(*A*, *B*). To specify both the edge to be linked and the endpoint of the edge used as the reference

point to locate points along the edge, we can use the associated half-edges to indicate the edges to be linked instead of the edges themselves.

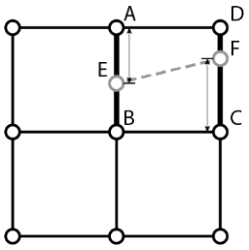

Figure 159. Parametric control of linking edge to edge.

8.6 Generative Algorithms

8.6.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock designs, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by X;
- the number of streets running horizontally in the initial regular grid, denoted by Y;
- the first edge to be linked (i.e., the edge to be linked from), as indicated by a half-edge h_1;
- the second edge to be linked (i.e., the edge to be linked to), as indicated by a half-edge h_2;

- the minimum ratio between the distance from the origin vertex of h_1 to the point to be linked along the edge $h_1.e$ and the length of the edge $h_1.e$, denoted by r_{min1};
- the maximum ratio between the distance from the origin vertex of h_1 to the point to be linked along the edge $h_1.e$ and the length of the edge $h_1.e$, denoted by r_{max1};
- the minimum ratio between the distance from the origin vertex of h_2 to the point to be linked along the edge $h_2.e$ and the length of the edge $h_2.e$, denoted by r_{min2};
- the maximum ratio between the distance from the origin vertex of h_2 to the point to be linked along the edge $h_2.e$ and the length of the edge $h_2.e$, denoted by r_{max2};
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled by the following parameters: l_x, l_y, X, Y, h_1, h_2, r_{min1}, r_{max1}, r_{min2}, r_{max2}, T.

8.6.2 General description

We start from a street graph that represents a regular grid. Then we randomly pick a cell c in the street graph. We further randomly pick two half-edges, h_1 and h_2, which belong to the cell c, and attempt to link the edge $h_1.e$ to the edge $h_2.e$. We continue to do so until the operation has been successfully performed a specified number of times. The operation can only be performed if *all* of the following conditions are met.

1. The area of the cell c is greater than or equal to half the area of the block in the initial street graph—that is, $A(c) \geqslant 0.5\, I_x I_y / [(X - 1)(Y - 1)]$.

2. The length of the first edge to be linked ($h_1.e$) is greater than half the length of the block in the initial street graph—that is, $L(h_1.e) > 0.5\ max\{l_x / (X - 1), l_y / (Y - 1)\}$.
3. The length of the second edge to be linked ($h_2.e$) is greater than half the length of the block in the initial street graph—that is, $L(h_2.e) > 0.5\ max\{l_x / (X - 1), l_y / (Y - 1)\}$.
4. Linking the new vertices created on the edges $h_1.e$ and $h_2.e$ meets all the conditions set for the operation of linking vertex to vertex as described in Chapter 6.

The first three conditions are included to prevent the generation of extremely small blocks and the occurrence of extremely short distances between intersections.

8.6.3 Pseudocode

Suppose that we have already generated the initial street graph G which represents a regular grid that is l_x units long, l_y units wide, with X number of streets running vertically and Y number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of linking edge to edge based on the initial street graph is described more precisely by the pseudocode shown below. In the pseudocode, *random* (a, b) refers to a procedure which returns a random floating-point number N such that $a \leqslant N \leqslant b$ for $a \leqslant b$, and *random_choice* (s) refers to a procedure which returns a random element from the set s.

ALGORITHM 8.2: Link edges to edges

```
procedure link_edges_to_edges (G, r_min1, r_max1, r_min2, r_max2, T)
counter ≔ 0
while counter < T
        c ≔ random_choice (C)
        if A(c) ≥ 0.5 l_x l_y / [(X − 1)(Y − 1)] then
                H_c ≔ a set comprising all half-edges that belong to c
                h_1 ≔ random_choice (H_c)
                if L(h_1.e) > 0.5 max{l_x / (X − 1), l_y / (Y − 1)} then
                        H_c' ≔ H_c − {h_1}
                        h_2 ≔ random_choice (H_c')
                        if L(h_2.e) > 0.5 max{l_x / (X − 1), l_y / (Y − 1)} then
                                r_1 ≔ random (r_min1, r_max1)
                                vec_h_1 ≔ (h_1.v.x − h_1.twin_h.v.x, h_1.v.y − h_1.twin_h.v.y)
                                p_1 ≔ (h_1.twin_h.v.x + r_1 * vec_h_1.x, h_1.twin_h.v.y + r *
                                vec_h_1.y)
                                r_2 ≔ random (r_min2, r_max2)
                                vec_h_2 ≔ (h_2.v.x − h_2.twin_h.v.x, h_2.v.y − h_2.twin_h.v.y)
                                p_2 ≔ (h_2.twin_h.v.x + r_2 * vec_h_2.x, h_2.twin_h.v.y + r *
                                vec_h_2.y)
                                perform the operation of linking p_1 to p_2
                                counter ≔ counter + 1
```

Note that the restrictions set for the operation of linking vertex to vertex also apply when we attempt to link the two vertices created along the existing edges. If the operation cannot be successfully performed, the points p_1 and p_2 will be removed, and the street graph will be restored to the state right before the current while-loop started.

8.7 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the six groups of designs, we varied only one control parameter—namely, T, the total number of operations to be performed—and kept all the others the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9, $r_{min1} = r_{max1} = r_{min2} = r_{max2} =$

0.5. By assigning the same value (0.5) to r_{min1}, r_{max1}, r_{min2}, and r_{max2}, we require that the operation always links the midpoint of the first edge to the midpoint of the second edge.

8.7.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation on the initial 9 × 9 square-grid design—that is, the values of T. We adopted the following strategy: First, we chose an arbitrarily high number a such that, based on our algorithm, it would be impossible to apply this operation more than a times on the initial design. Then we ran the algorithm by setting $T = a$ and applied the following terminating condition: If 50000 attempts have been made in a row and none was successful, then stop making any more attempts and record the total number of times that the operation has been successfully performed—let's denote it by b. It gave us a sense of the upper limit for the number of successful applications of the operation for a particular run. Notice that the number b may vary for different runs, thus we made 100 trial runs and recorded the value of b after each trial. Then we chose the smallest recorded value as the maximum number of times to apply the operation in the generative process—let's denote it by T_{max}. The minimum number of times to apply the operation, T_{min}, was subsequently determined by finding the nearest integer for $T_{max} / 6$. To generate the other four groups of designs, we simply set T to $2T_{min}$, $3T_{min}$, $4T_{min}$, and $5T_{min}$, respectively.

Examples of designs from each group are shown in Figure 160.

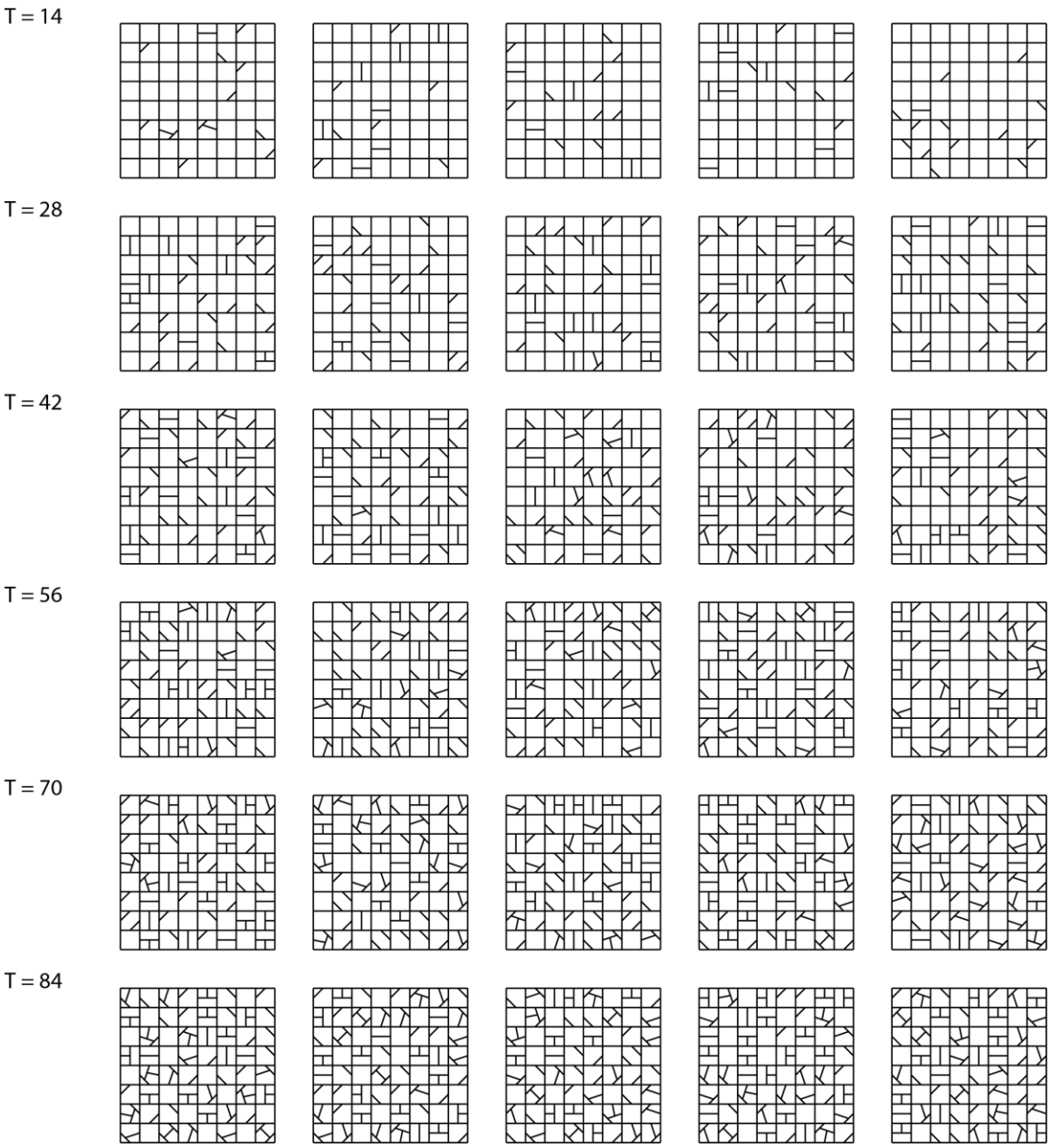

Figure 160. Examples of designs from each group (link edge to edge).

8.7.2 Data analysis

Six groups of designs were generated by applying the operation of linking edge to edge 14, 28, 42, 56, 70, and 84 times, respectively, on the initial 9 × 9 square-grid design.

Each group consists of 100 designs. The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of designs.

Elementary graph properties

(1) Number of vertices

By definition, each application of the operation creates two new vertices—one on each edge to be linked. Therefore, as shown in Figure 161, the total number of vertices per design increases as the operation is applied more frequently. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the total number of vertices, $|V|$, equals $81 + 2T$.

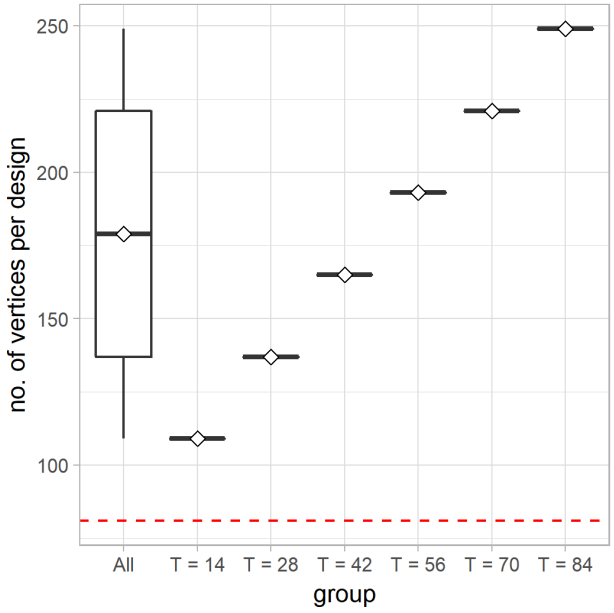

Figure 161. A boxplot (with mean diamonds) showing the distribution of the total number of vertices per design for all designs and for each group of designs (link edge to edge). The red dashed line indicates the total number of vertices in the initial 9 × 9 square-grid design (81).

(2) Number of edges

Each time we apply the operation of linking edge to edge, we create two new vertices on the edges to be linked—one on each edge. By placing a new vertex along an existing edge, we split the edge into two edges. Therefore, placing two new vertices on the two existing edges results in four edges. Since we also add an additional edge to connect the two newly created vertices, we increase the total number of edges by three after each application of the operation of linking edge to edge. As shown in Figure 162, the total number of edges per design increases rapidly as the operation is applied more frequently. More specifically, for a design generated by applying the operation of linking edge to edge T times on the initial 9 × 9 square-grid design, the total number of edges, $|E|$, equals $144 + 3T$.

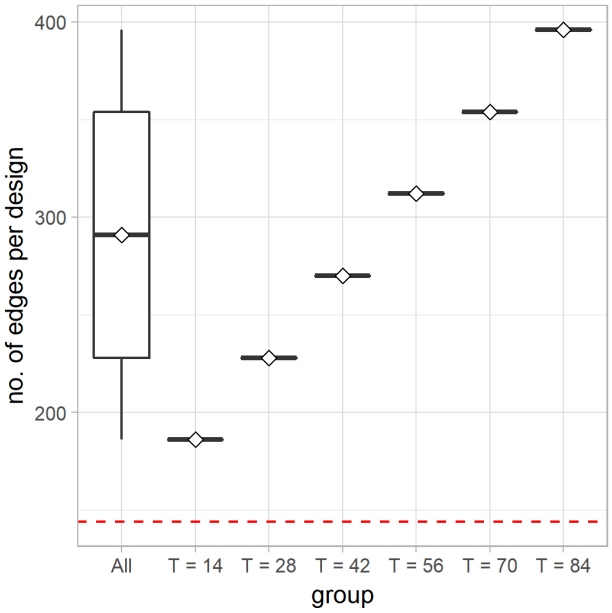

Figure 162. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (link edge to edge). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

By definition, each time we apply the operation of linking edge to edge, we split an existing cell into two, thus increasing the total number of cells by one. As shown in Figure 163, the total number of cells per design increases steadily as the operation is applied more frequently. More specifically, for a design generated by applying the operation of linking edge to edge T times on the initial 9 × 9 square-grid design, the total number of cells, $|C|$, equals $64 + T$.

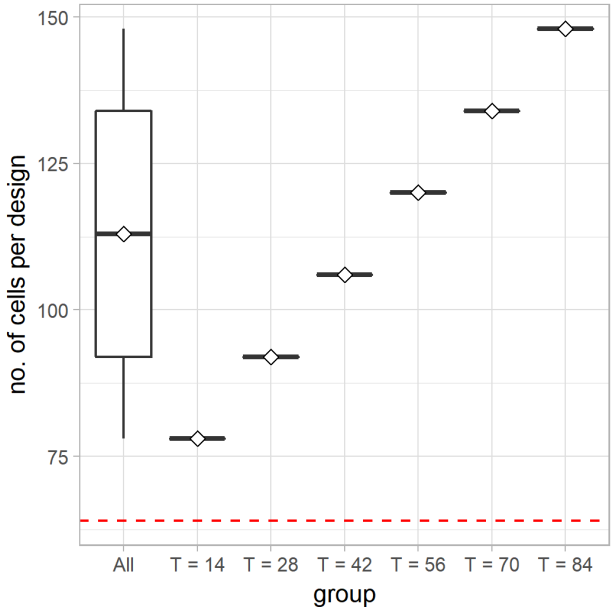

Figure 163. A boxplot (with mean diamonds) showing the distribution of the total number of cells per design for all designs and for each group of designs (link edge to edge). The red dashed line indicates the total number of cells in the initial 9 × 9 square-grid design (64).

(4) Vertex degree

Each application of the operation of linking edge to edge contributes two vertices of degree 3, or two T-intersections. Therefore, just as in the initial square-grid design, a vertex in a design generated by our algorithm can have degree 2, 3, or 4. As the operation is applied more frequently, more vertices of degree 3 are generated, making the mean vertex degree per design closer to 3.

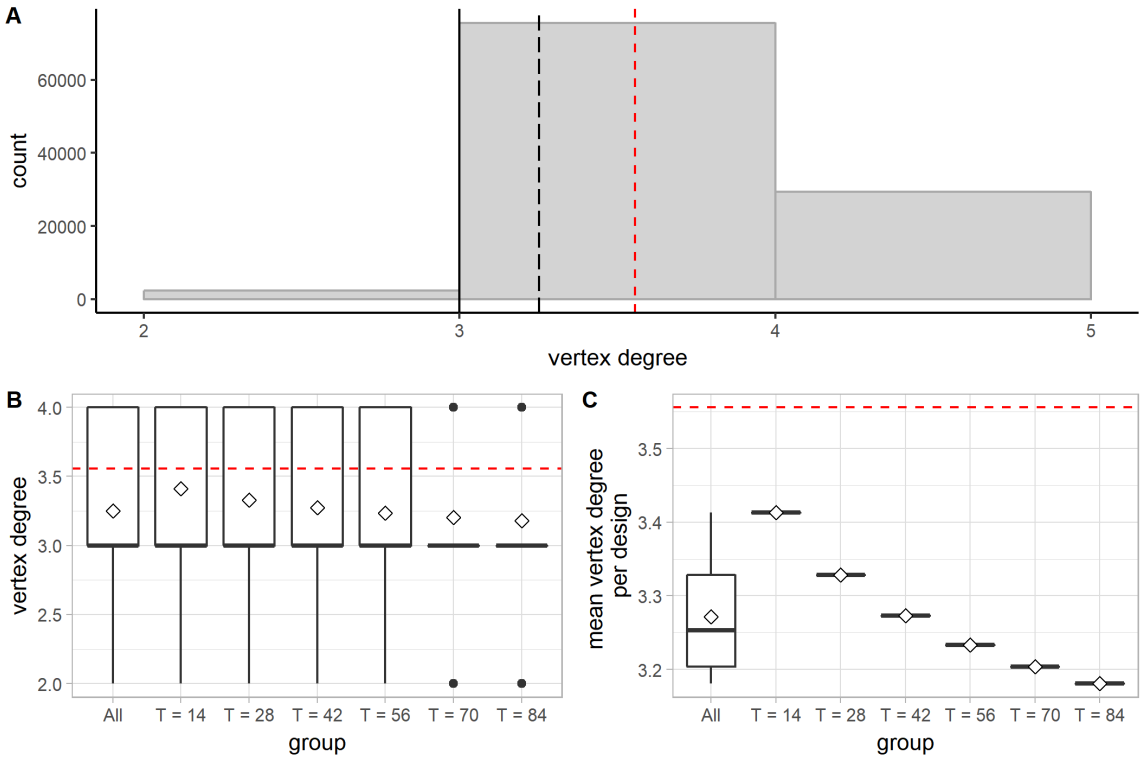

Figure 164. Distribution of the vertex degree (link edge to edge). A: A histogram showing the distribution of the vertex degree for all vertices in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design ($\approx$ 3.556). B: A boxplot (with mean diamonds) showing the distribution of the vertex degree for all vertices and for vertices in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs.

As shown in Figure 164C, when analyzed in groups, as *T* increases, the mean vertex degree per design decreases and tends to 3. Based on the handshaking theorem, the sum of the degrees of the vertices of a graph is twice the number of its edges. Therefore, the mean vertex degree for a graph is $2|E| / |V|$. For a design generated by applying the operation *T* times on the initial 9 × 9 square-grid design, the mean vertex degree equals $2(144 + 3T) / (81 + 2T)$.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

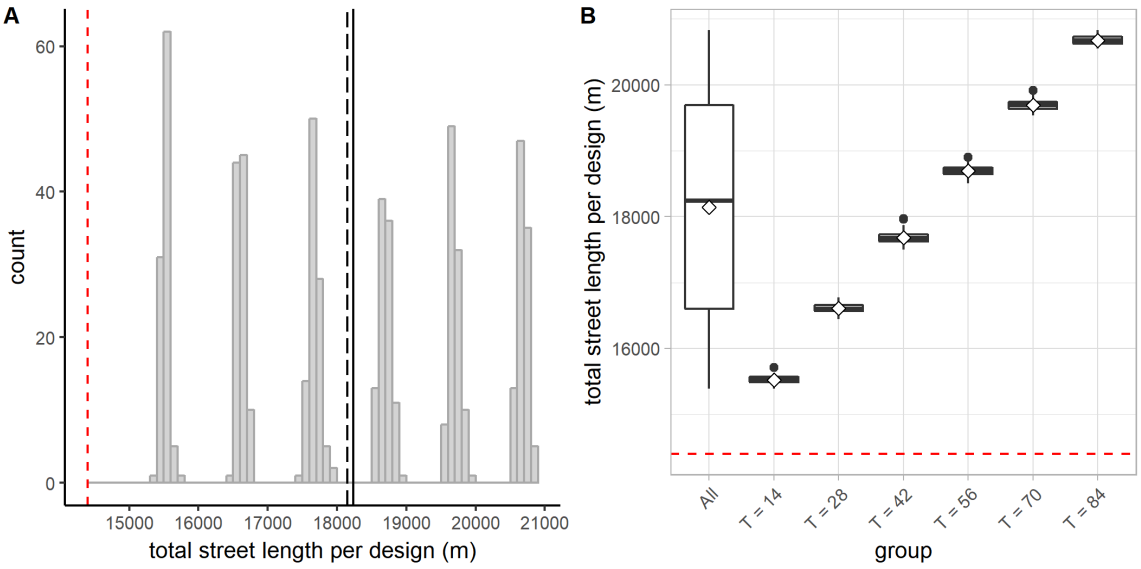

Figure 165. Distribution of the total street length per design (link edge to edge). A: A histogram showing the distribution of the total street length per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs.

Each time we apply the operation of linking edge to edge, we add an additional street, thus increasing the total street length in the design. As shown in Figure 165B, the total street length per design clearly tends to increase as the operation is applied more frequently.

(2) Total number of blocks per design

The total number of blocks per design is the same as the total number of cells per design. It increases as the operation is applied more frequently.

(3) Total number of intersections per design

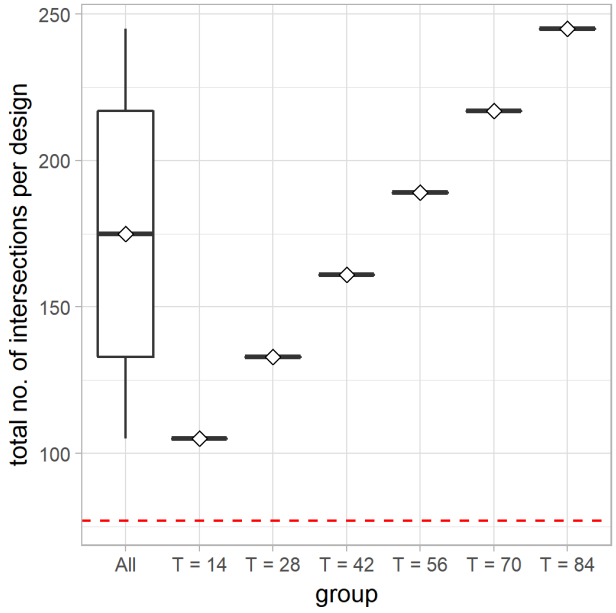

Figure 166. A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs (link edge to edge). The red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77).

Since each application of the operation of linking edge to edge contributes two T-intersections, the total number of intersections per design increases steadily as the

operation is applied more frequently (Figure 166). More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the total number of intersections is $77 + 2T$.

(4) Distance between intersections

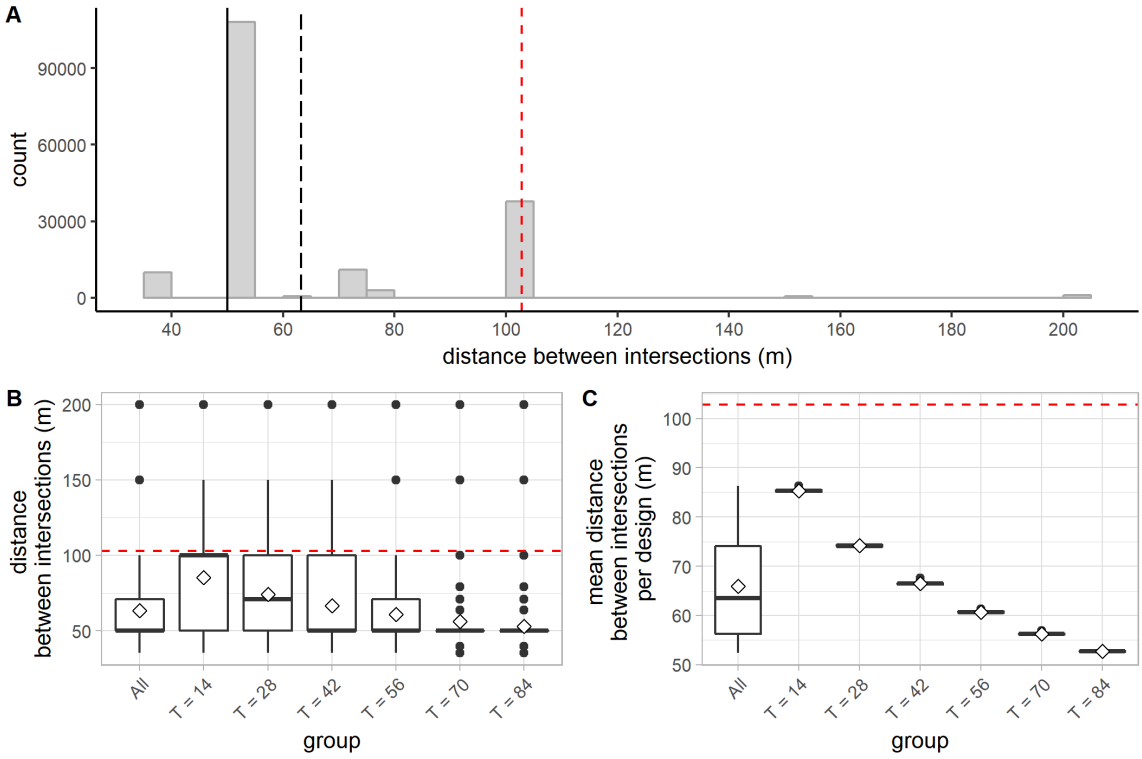

Figure 167. Distribution of the distance between intersections (link edge to edge). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the distance between intersections observed in all designs and in each group of designs. C: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Based on our algorithm and the setup of the initial street graph, the distance between intersections can only take one of nine distinct values—all of them are smaller

than the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). By splitting the edges to be linked, the operation breaks long segments into shorter segments, thus tends to decrease the distances between intersections. As shown in Figure 167C, the mean distance between intersections per design tends to decrease as the operation is applied more frequently.

Directional reach and directional distance

(1) DDL

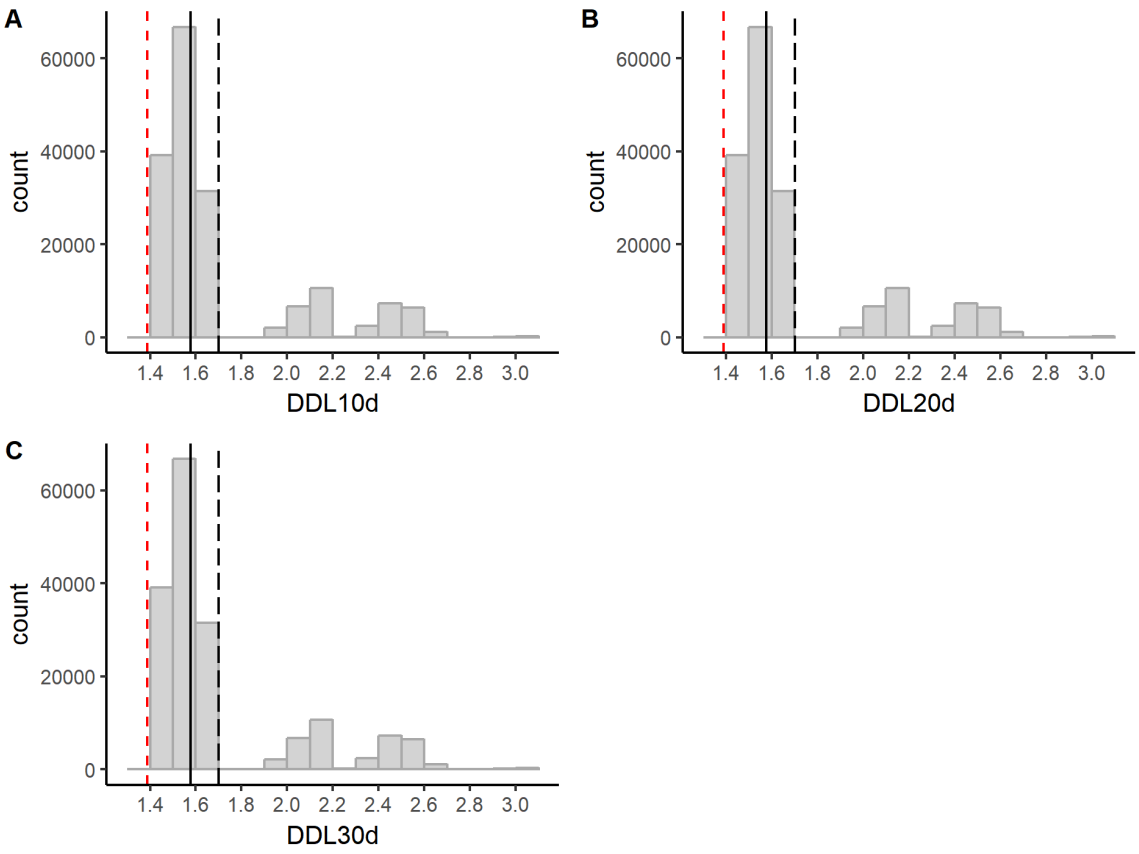

Figure 168. Distributions of DDL10d, DDL20d, and DDL30d (link edge to edge). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

Based on our algorithm and the setup of the initial street graph, the angle of deviation involved in transitioning between segments would never be between 10° and 30°. Therefore, the distribution of DDL looks identical as the threshold angle is increased from 10° to 30° (Figure 168). To be consistent with other chapters, in the following analysis, unless otherwise specified, the directional distance is always evaluated by setting the threshold angle to 20°.

For all the segments analyzed, DDL20d ranges from 1.40 to 3.06. Nearly 80% of all segments assume a DDL20d less than 1.65. The lower peaks in the histogram as shown in Figure 168B are attributed to the segments added inside the square blocks in the initial square-grid design. On average, they are about one more turns away from the rest of the system than the horizontal and vertical streets in the initial street system. The even more segregated segments are the ones that are not attached to the main grid—in other words, they are internal streets nested inside the original square blocks. Here, we start to see the emergence of scales/levels based on their syntactic properties. As shown in Figure 169A, the DDL20d values seem to be separated into different "layers", suggesting a form of "syntactic stratification". When analyzed in groups, as *T* increases, the mean DDL20d per design tends to increase, as shown in Figure 169B.

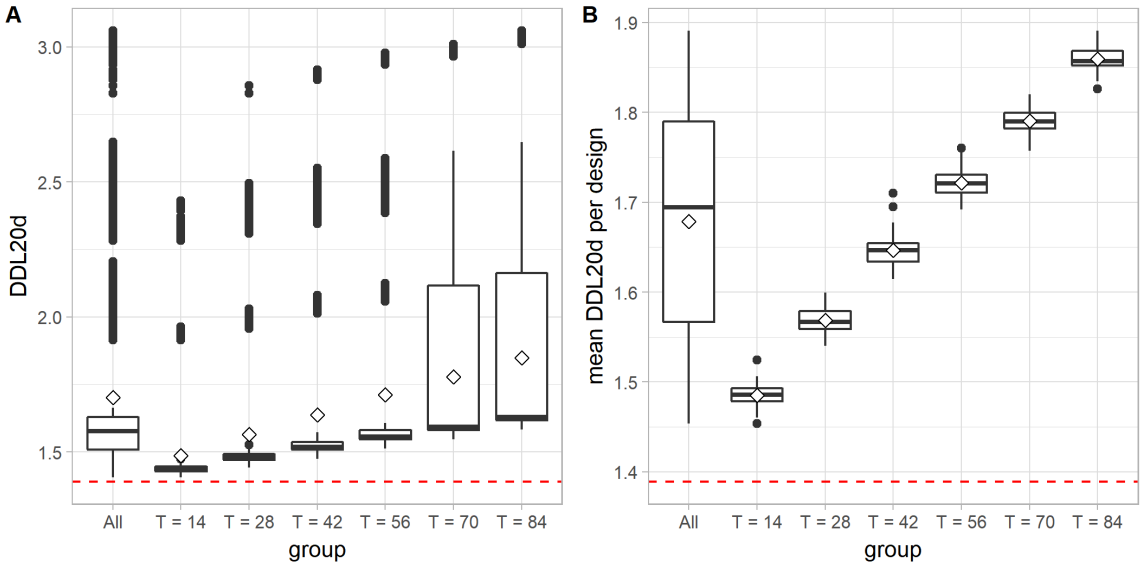

Figure 169. Distribution of DDL20d (link edge to edge). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 170, most segments have a linear reach of 800 m—they are the segments forming the orthogonal grid in the initial square-grid design. The segments which have a linear reach less than or equal to 100 m are those added later in the generative process. As shown in Figure 170C, the mean linear reach per design tends to decrease as the operation is applied more frequently.

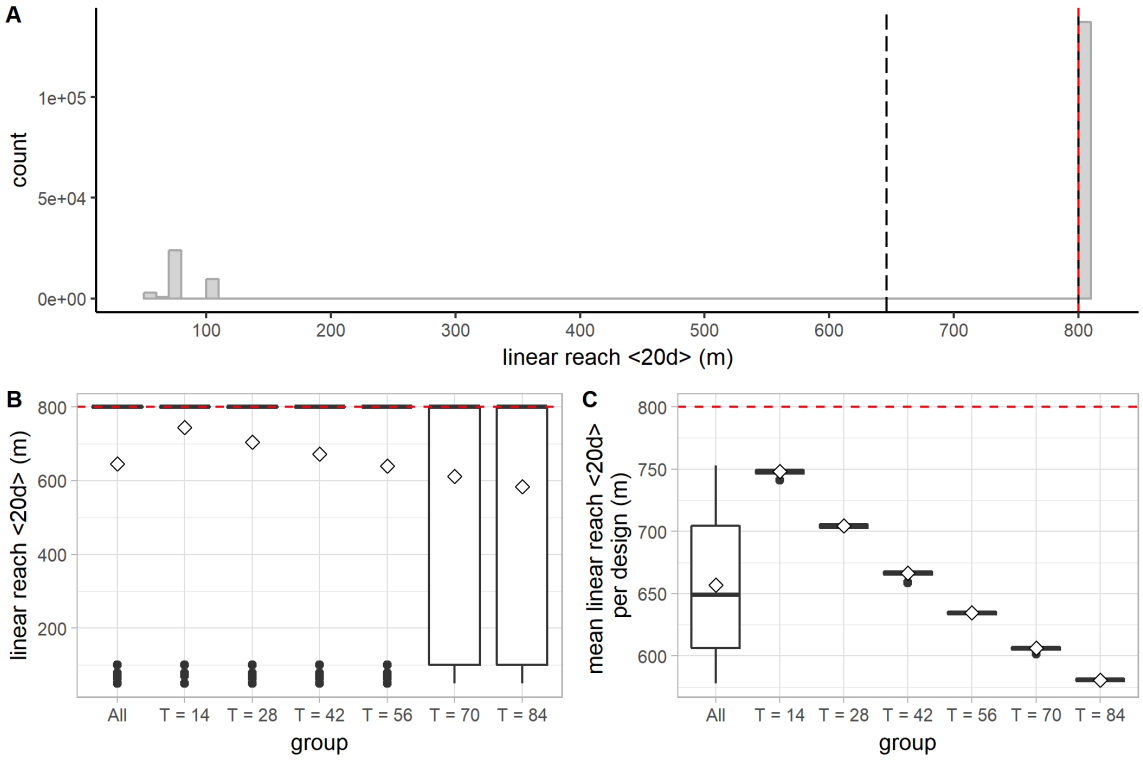

Figure 170. Distribution of the linear reach (link edge to edge). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

(3) 2-dc reach (dr2dc20d)

As shown in Figure 171, most segments assume a 2-dc reach value greater than 14400 m—that is, the total street length for the initial square-grid design. This is not surprising because, for all the segments lying on the main grid (i.e., the street network formed by the horizontal and vertical streets in the initial square-grid design) as well as the segments added later which are connected to both a horizontal and a vertical street in the main grid, they can reach all the streets lying on the main grid within two direction

changes. The streets added later inside the square blocks which are connected to two parallel streets in the main grid can only reach about half length of the main grid within two direction changes (corresponding to the middle cluster in the histogram shown in Figure 171A), while the internal streets which do not have direct access to the main grid can only reach a small portion of the main grid within two direction changes.

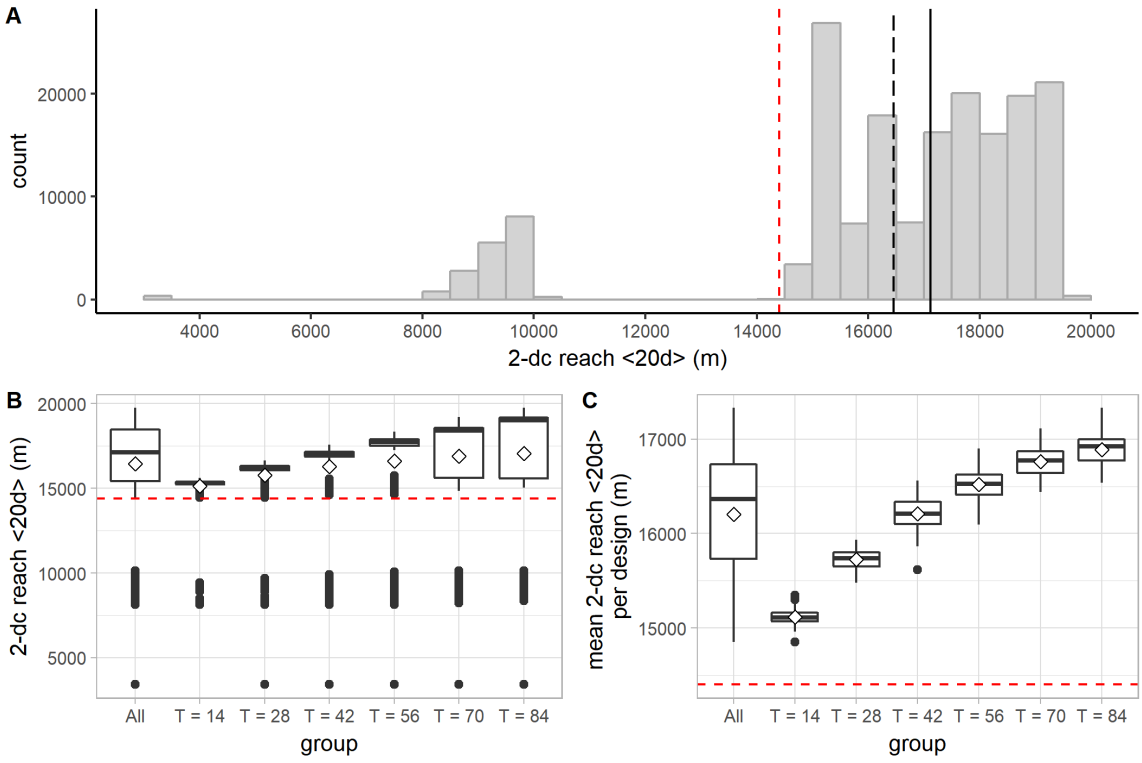

Figure 171. Distribution of the 2-dc reach (link edge to edge). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

The mean 2-dc reach per design depends on two factors: the density of the street network as well as the alignment of streets. As shown in Figure 171C, the mean 2-dc reach per

design tends to increase as the operation is applied more frequently, mainly because of the increasing density of the street network.

Geometric regularity

(1) Fragmentality per design

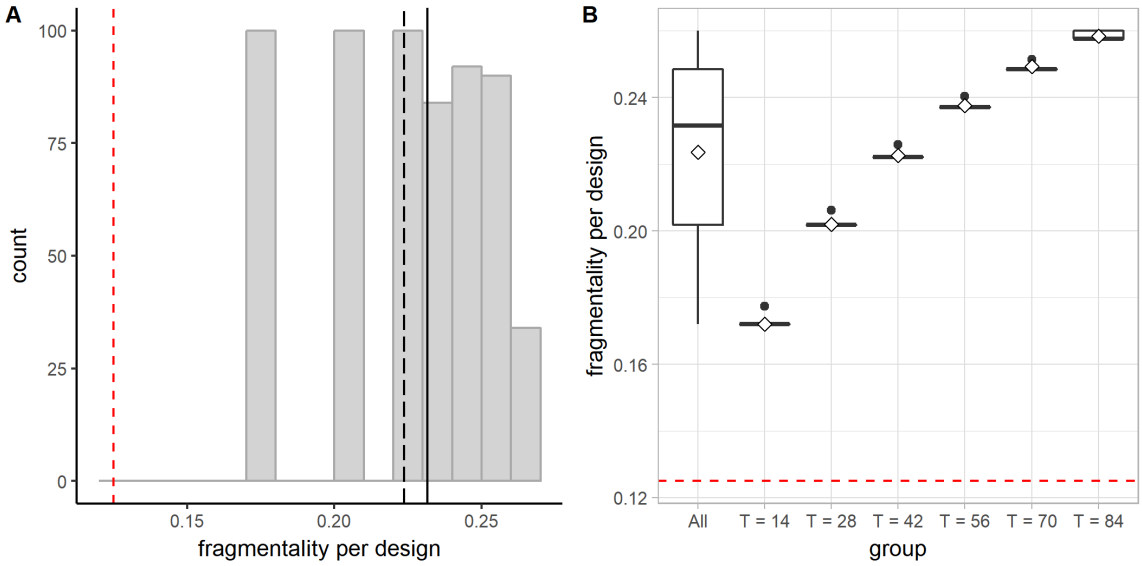

Figure 172. Distribution of the fragmentality per design (link edge to edge). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

The fragmentality per design ranges from 0.17 to 0.26, and all the designs analyzed are more fragmented than the initial square-grid design. This is not surprising because the segments added later in the generative process are not aligned linearly with other segments, thus impossible to form long continuity lines. As shown in Figure 172B,

when analyzed in groups, as *T* increases, the fragmentality per design tends to increase as well.

(2) Block area

Based on our algorithm and the setup of the initial street graph, the block area can only take one of eight different values (Figure 173).

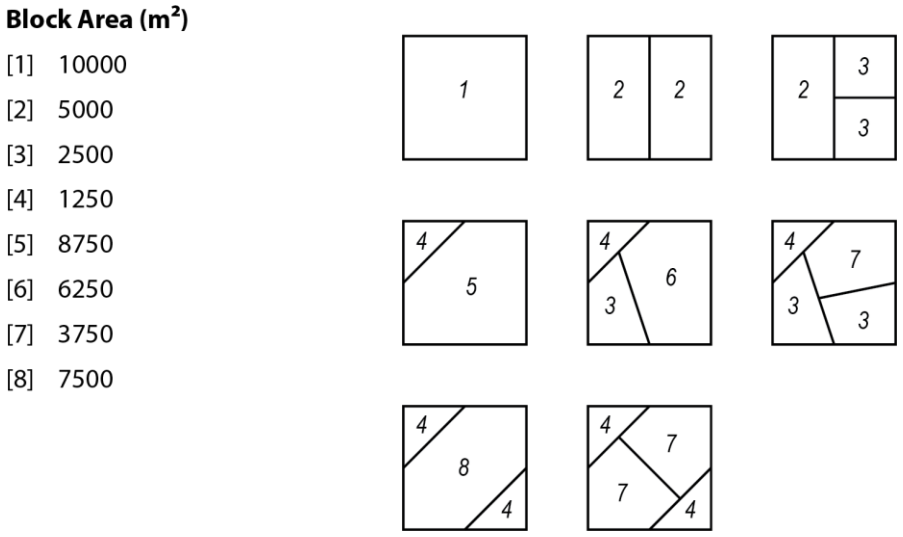

Figure 173. Enumerate all possible block sizes in designs generated by our algorithm (link edge to edge).

Obviously, as the operation is applied more frequently, the blocks are further subdivided, thus the mean block area per design is reduced (Figure 174C). More specifically, for a design generated by applying the operation of linking edge to edge *T* times on the initial 9 × 9 square-grid design, the mean block area per design is 800 × 800 / (64 + *T*) m², or 640000 / (64 + *T*) m².

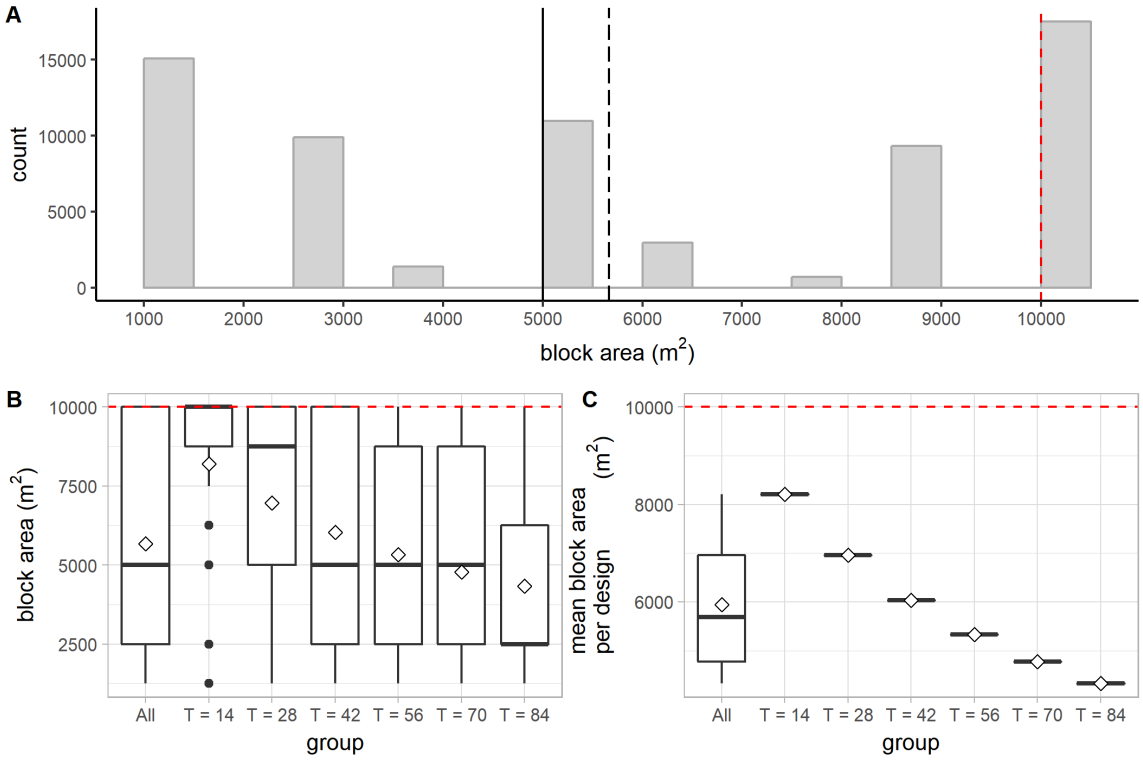

Figure 174. Distribution of the block area (link edge to edge). A: A histogram showing the distribution of the area for all the blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²). B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs. C: A boxplot showing the distribution of the mean block area per design for all designs and for each group of designs.

As shown in Figure 175A, initially, the standard deviation of the block area per design tends to increase, but after the operation has been applied 40 times or so, a downward trend appears to have taken place. To factor in the size of the mean block area, we plotted the coefficient of variation (CV) of the block area per design for each group of designs. Generally, it shows that the CV of the block area per design tends to increase as the operation is applied more frequently (Figure 175B).

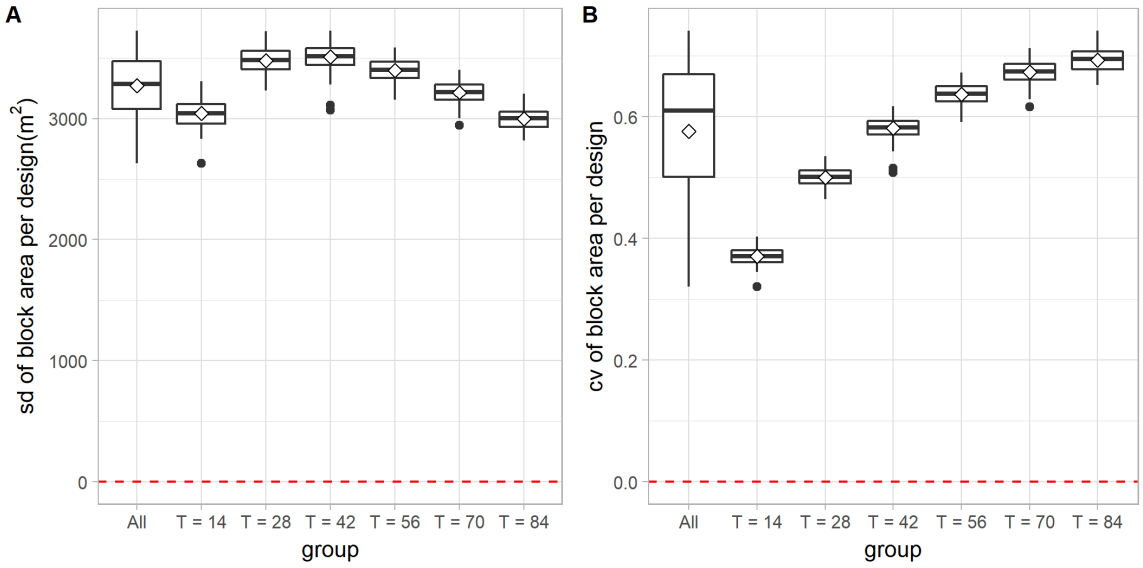

Figure 175. Variation of the block area per design (link edge to edge). A: A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs. The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²). B: A boxplot showing the distribution of the coefficient of variation (CV) of the block area per design for all designs and for each group of designs. The red dashed line indicates the CV of the block area per design for the initial 9 × 9 square-grid design (0.0).

(3) Block perimeter

Based on our algorithm and the setup of the initial street graph, the block perimeter can only take one of 11 different values, corresponding to the 11 different shapes possibly taken by a block (Figure 176).

Block Perimeter (m)

[1] 400

[2] 300

[3] 200

[4] $100 + 50\sqrt{2}$

[5] $300 + 50\sqrt{2}$

[6] $100 + 25(\sqrt{2} + \sqrt{10})$

[7] $200 + 25(\sqrt{2} + \sqrt{10})$

[8] $100 + 25[\sqrt{(5/2)} + \sqrt{(13/2)}]$

[9] $100 + 25[\sqrt{2} + \sqrt{(5/2)} + \sqrt{(13/2)}]$

[10] $200 + 100\sqrt{2}$

[11] $100 + 100\sqrt{2}$

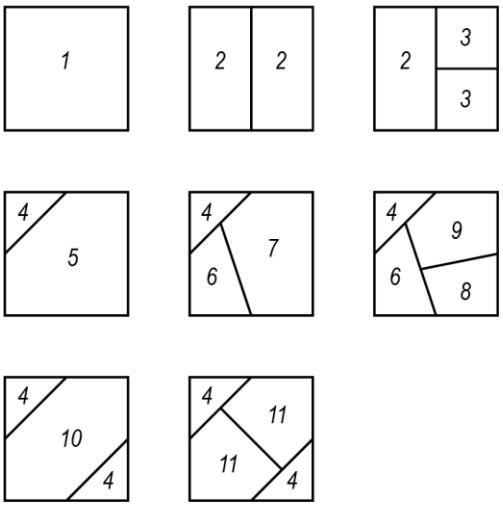

Figure 176. Enumerate all possible block perimeters in designs generated by our algorithm (link edge to edge).

As the operation is applied more frequently, the blocks are subdivided into smaller blocks. Therefore, when analyzed in groups, as T increases, the mean block perimeter per design tends to be reduced, as shown in Figure 177C.

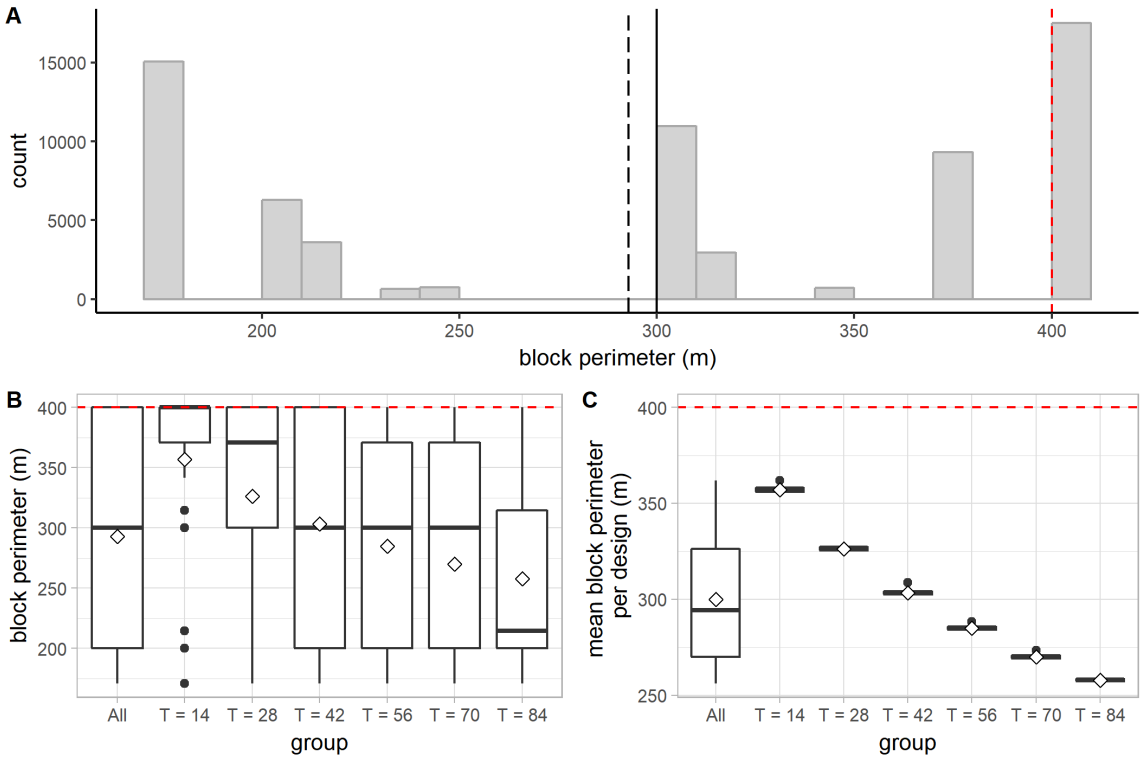

Figure 177. Distribution of the block perimeter (link edge to edge). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

As shown in Figure 178A, as the operation is applied more frequently, the standard deviation of the block perimeter per design tends to increase at first. However, after the operation has been applied 40 times or so, it starts to decrease. To factor in the diminishing mean block perimeter, we plotted the coefficient of variation (CV) of the block perimeter per design for each group of designs. As shown in Figure 178B, when analyzed in groups, the CV of the block perimeter per design tends to increase until *T* reaches 70. After that point, the upward trend disappears.

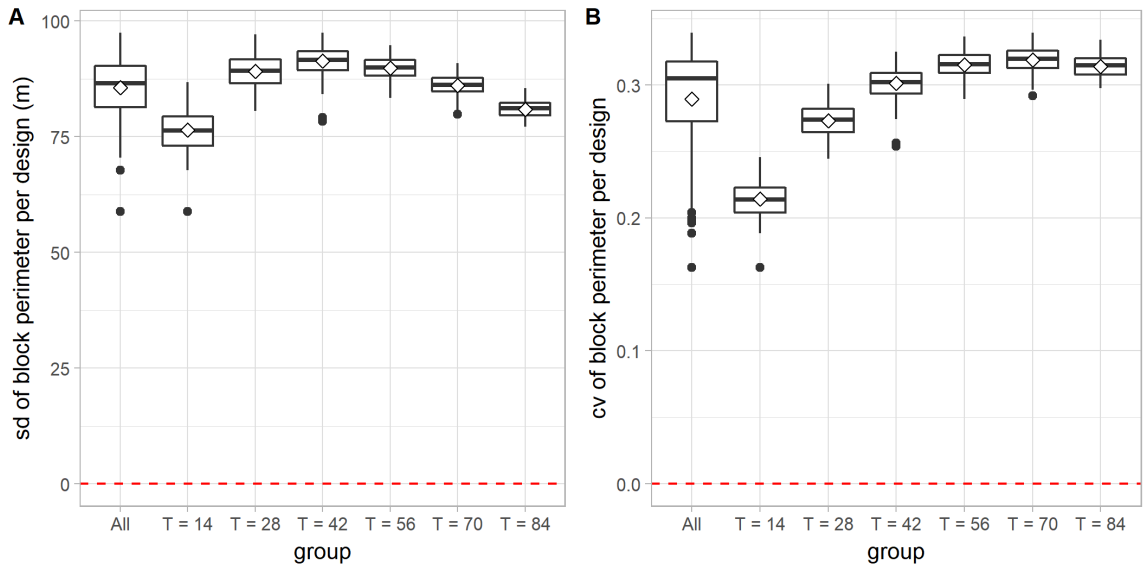

Figure 178. Variation of the block perimeter per design (link edge to edge). A: A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs. The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m). B: A boxplot showing the distribution of the coefficient of variation (CV) of the block perimeter per design for all designs and for each group of designs. The red dashed line indicates the CV of the block perimeter per design for the initial 9 × 9 square-grid design (0.0).

(4) Standardized block area-perimeter ratio (SAPR)

Based on our algorithm and the setup of the initial street graph, the standardized block area-perimeter ratio can only take one of nine different values (Figure 179).

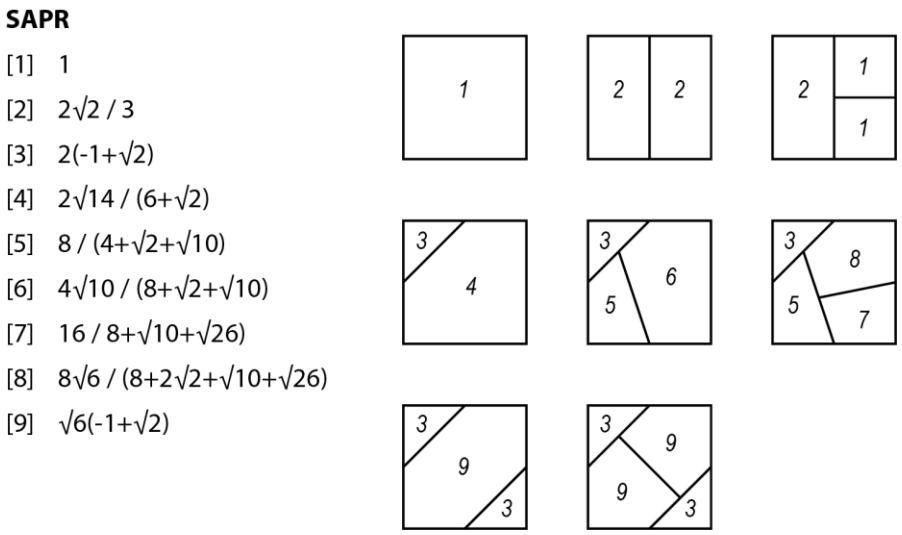

Figure 179. Enumerate all possible SAPR values taken by blocks in designs generated by our algorithm (link edge to edge).

For all the blocks analyzed, the triangular blocks resulted from the subdivision have the least SAPR (≈ 0.83), while the blocks that have more than four sides have SAPR values greater than 1—in a sense, they are more compact than the perfect square. As shown in Figure 180C, when analyzed in groups, as *T* increases, the mean SAPR per design tends to decrease. The downward trend disappears after the operation has been applied 50 times or so.

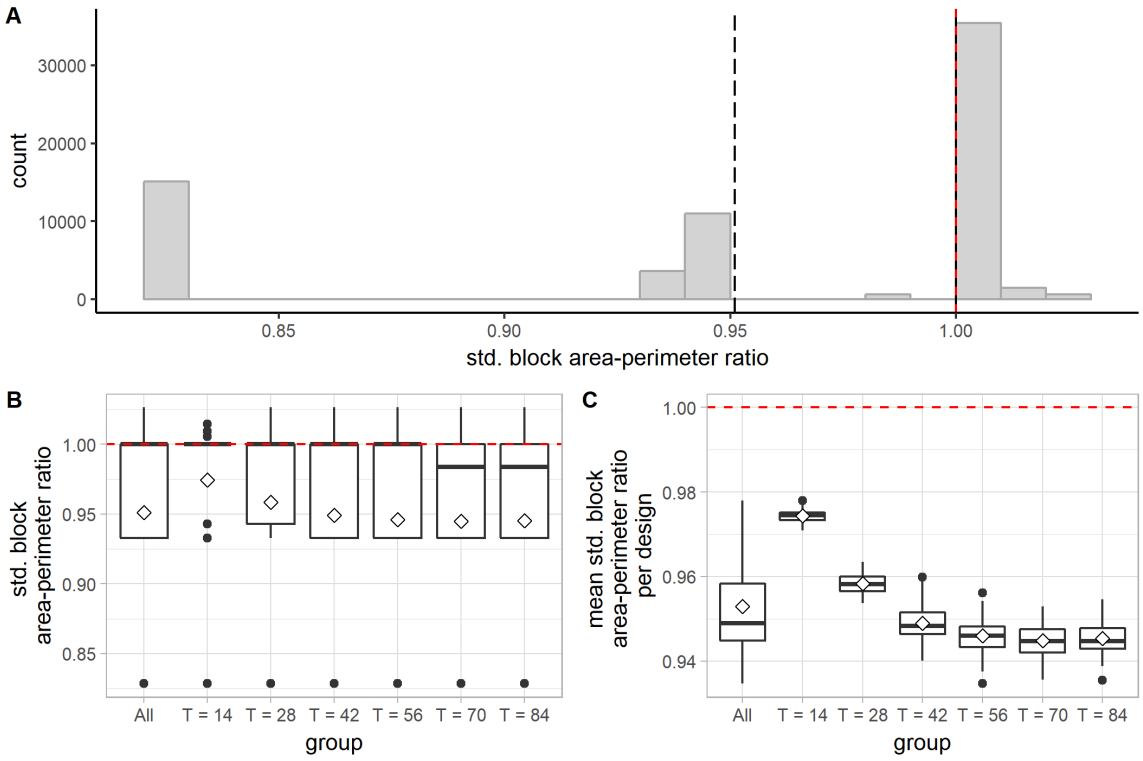

Figure 180. Distribution of the standardized block area-perimeter ratio (SAPR) (link edge to edge). A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and proportion of distinct DDL20d values per design

As shown in Figure 181, when analyzed in groups, the total number of distinct DDL20d values per design clearly tends to increase as *T* increases. Considering that the total number of segments/edges increases as well during the generative process, we also plotted the proportion of distinct DDL20d values per design to counteract the effect of the

growing size of the street graph. The proportion of distinct DDL20d values per design ranges from 0.07 to 0.26—a relatively narrow range. As shown in Figure 181C, the proportion of distinct DDL20d values per design also tends to increase as *T* increases.

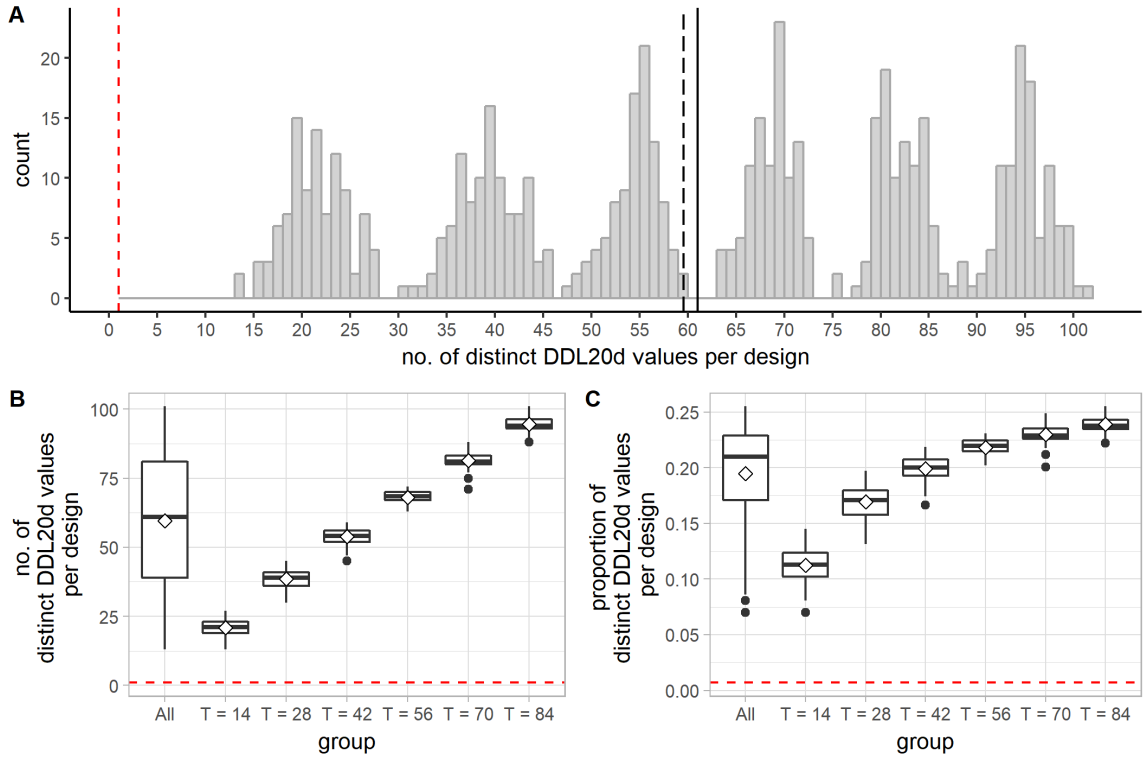

Figure 181. Distribution of the total number and the proportion of distinct DDL20d values per design (link edge to edge). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d values per design

For all the designs analyzed, the standard deviation of DDL20d per design ranges from 0.13 to 0.40. As shown in Figure 182B, when analyzed in groups, as *T* increases, the standard deviation of DDL20d per design tends to increase as well.

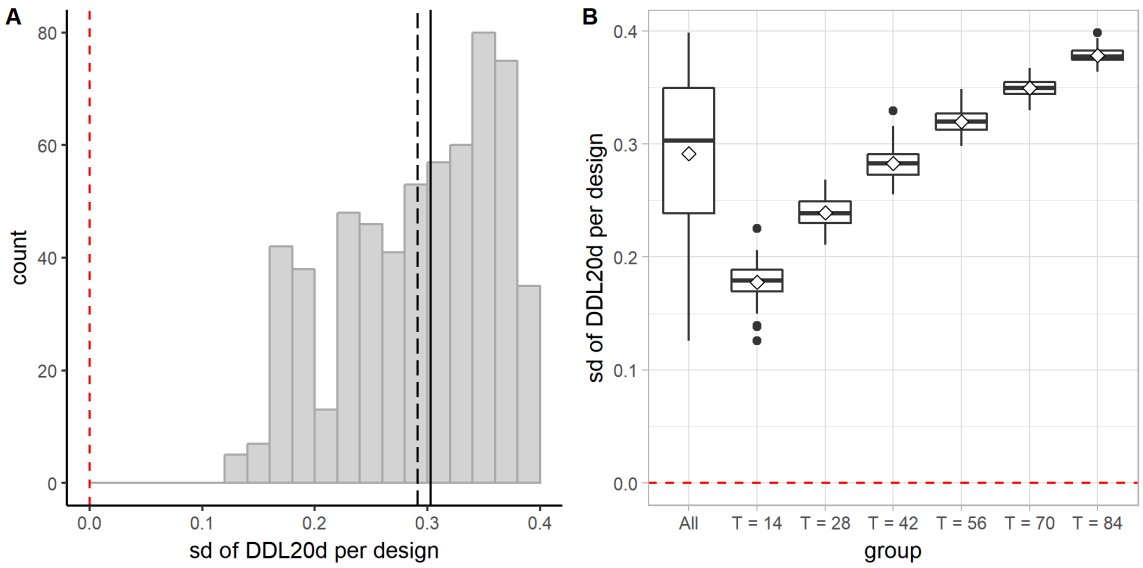

Figure 182. Distribution of the standard deviation of DDL20d per design (link edge to edge). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

8.8 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: Each application of the operation increases the total number of vertices by two, the total number of edges by three, and the total number of cells by one. The operation always creates vertices of degree 3 and tends to reduce the mean vertex degree per design (but not by much compared to the mean vertex degree of the initial square-grid design).

The impact of the operation on the density of a street network: The operation increases the total street length per design, the total number of blocks per design, as well as the total number of intersections per design. By breaking long segments into shorter segments, this operation tends to decrease the mean distance between intersections per design.

The impact of the operation on the directional reach/distance: The operation tends to increase the mean DDL20d per design because the segments added inside the original square blocks are more secluded from other places compared to the initial main grid. The DDL20d values can be separated into a few clusters, suggesting a hierarchy of streets in terms of syntactic conditions. The operation tends to decrease the mean linear reach per design because the newly added segments are not aligned linearly with existing streets. It tends to increase the mean 2-dc reach per design mainly because of the increasing density of the street network, which overcomes the effect of the misaligned street segments.

The impact of the operation on the regularity of a street network: The operation tends to increase the fragmentality per design. Both the mean block area and the mean block perimeter per design get smaller as the operation is applied more frequently. The blocks also tend to be more varied in terms of their size and perimeter—especially when

the operation has not been applied too many times, as evidenced by the increasing coefficients of variation. As the operation is applied more frequently, the mean standardized area-perimeter ratio per design tends to decrease, but it soon stabilizes after the operation has been applied 50 times or so.

The impact of the operation on the diversity of syntactic conditions: The operation tends to increase the diversity of syntactic conditions in a superblock design. However, there is generally a lack of diversity in syntactic conditions, as evidenced by the low proportion of distinct DDL20d values per design. Nevertheless, the syntactic conditions tend to be more varied for designs generated by more frequent application of this operation, as evidenced by the increasing standard deviation of DDL20d values per design.

CHAPTER 9
SPLITTING VERTEX: CREATING INTERRUPTED GRIDS

9.1 Operation: Split Vertex

9.1.1 Definition of operation

The operation of splitting vertex splits an existing vertex in two and pulls them apart by "swinging" one of the incident edges along another incident edge that lies to the immediate left (which would be a "left-split" with respect to the edge being swung) or right (which would be a "right-split" with respect to the edge being swung) side of it (Figure 183).

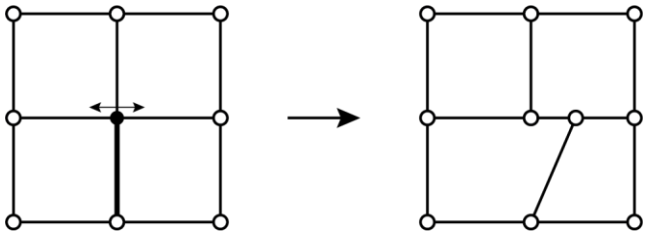

Figure 183. Split a vertex.

9.1.2 Parameters

To precisely control the operation of splitting vertex, we need to specify in which direction and how far to pull the vertices apart. For example, suppose that we are splitting the vertex *A* as shown in Figure 184. First, we need to specify which edge we are going to swing. Assume that we are swinging the edge {*B*, *A*}, then the next step is to specify in which direction we are swinging the edge. Assume that we are swinging the edge {*B*, *A*} to the right along the edge {*A*, *D*}—which would be a "right-split" with respect to {*A*, *B*},

we still need to specify the distance between the vertex *A* and the vertex *C*—the new vertex resulted from the splitting operation, denoted by *d*(*A*, *C*). To fully specify how the operation is to be performed, we can use a half-edge, *h*, to indicate both the vertex to be split (which would be *h.v*) and the edge to be swung (which would be *h.e*), in addition to specifying in which direction to swing the edge and how far to swing the edge/pull the vertices apart.

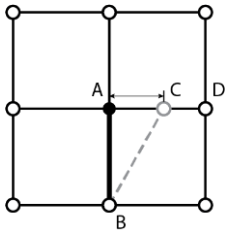

Figure 184. Parametric control of splitting vertex.

9.2 Generative Algorithms

9.2.1 Control parameters

The generative process begins with a street graph that represents a regular grid. Before we describe the algorithm developed to generate the superblock designs, we introduce the parameters used to control the generative process. They include

- the length of the initial regular grid, denoted by l_x;
- the width of the initial regular grid, denoted by l_y;
- the number of streets running vertically in the initial regular grid, denoted by *X*;
- the number of streets running horizontally in the initial regular grid, denoted by *Y*;
- the vertex to be split, as indicated by a half-edge *h*;

- the minimum degree of the vertex must have to qualify for this operation, denoted by τ;
- the minimum distance between the newly split vertex and either endpoint of the edge to be swung along, denoted by d_{min};
- the maximum distance between the split vertices, denoted by d_{max};
- the total number of times the operation should be performed, denoted by T.

To summarize in symbols, the generative process can be parametrically controlled by the following parameters: l_x, l_y, X, Y, h, τ, d_{min}, d_{max}, T.

9.2.2 General description

We start from a street graph that represents a regular grid. Then we randomly pick a vertex, say v, from the street graph, and get all the half-edges that points to v. We further randomly pick a half-edge, say h, among all the half-edges pointing to v and then attempt to perform either a right-split or a left-split based on the selected half-edge. We continue to do so until the operation has been successfully performed a specified number of times. The operation can only be performed if *all* of the following conditions are met:

1. The vertex to be split has degree greater than the minimum degree required for this operation. This condition can be simply expressed as $deg(v) \geqslant \tau$.
2. The selected half-edge h (which indicates the edge to be swung) does not lie on the border of the street graph.
3. The edge $h.e$ is not a dangling edge.

4. If it is a left-split, then the length of *h.next_h.e* is greater than or equal to $2d_{min}$—otherwise the minimum distance requirement cannot be possibly satisfied.
5. If it is a left-split, then the path formed by the three vertices—*h.twin_h.v*, *h.v*, *h.next_h.v*—indicates a left-turn and the turning angle is not smaller than 5°.
6. If it is a right-split, then the length of *h.twin_h.prev_h.e* is greater than or equal to $2d_{min}$—otherwise the minimum distance requirement cannot be possibly satisfied.
7. If it is a right-split, then the path formed by the three vertices—*h.twin_h.v*, *h.v*, *h.twin_h.prev_h.twin_h.v*—indicates a right-turn and the turning angle is not smaller than 5°.
8. The edge, after being swung to the new position, does not lead to any unexpected intersections with other edges.

Below illustrate a few situations where some of the above conditions are not met.

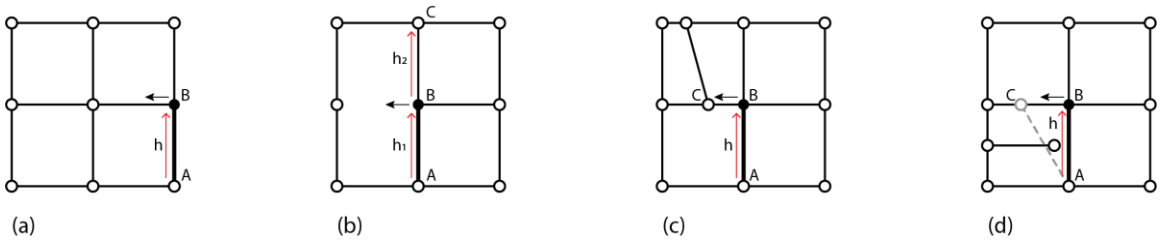

Figure 185. Examples of illegitimate moves (split vertex).

In the example shown in Figure 185a, the half-edge *h* involved in the operation lies on the border of the street graph—that is, a violation of Condition 2. Assume that the minimum degree the vertex must have to qualify the operation is 4 (i.e., $\tau = 4$), then it would also violate Condition 1 because the vertex *B* has degree 3.

In the example shown in Figure 185b, suppose that we are attempting to perform a left-split of the vertex *B* with respect to the half-edge h_1. The operation cannot be performed because the path formed by the vertices *A*, *B*, and *C* does not indicate a left-turn (instead, it is a straight path)—that is, a violation of Condition 5.

In the example shown in Figure 185c, suppose that we are attempting to perform a left-split of the vertex *B* with respect to the half-edge *h*. However, if the minimum distance (d_{min}) required between the newly split vertex and either end of the edge {*B*, *C*} is greater than half the length of {*B*, *C*}, then the operation cannot be successfully performed because of the violation of Condition 4.

In the example shown in Figure 185d, the intended left-split of the vertex *B* with respect to the half-edge *h* cannot be successfully performed because it would intersect the dangling edge—that is, a violation of Condition 8.

9.2.3 Pseudocode

Suppose that we have already generated the initial street graph *G* which represents a regular grid that is l_x units long, l_y units wide, with *X* number of streets running vertically and *Y* number of streets running horizontally. The algorithm developed to generate the superblock designs with the operation of splitting vertex based on the initial street graph is described more precisely by the pseudocode shown below.

ALGORITHM 9.1: Split vertices

```
procedure split_vertices (G, τ, d_min, d_max, T)
counter ≔ 0
while counter < T
        v ≔ random_choice (V)
        if deg(v) ≥ τ then
                H ≔ a set consisting of all the half-edges that point to v
                h ≔ random_choice (H)
                if h does not lie on the border of the street graph G
                and h.e is not a dangling edge then
                        dir ≔ random_choice ({"left", "right"})
                        /* left-split */
                        if dir == "left" then
                                h_left ≔ h.next_h
                                if L(h_left.e) ≥ 2d_min
                                and is_left_turn (h.twin_h.v, h.v, h_left.v)
                                and dev_angle (h.twin_h.v, h.v, h_left.v) ≥ 5° then
                                        d ≔ random (d_min, d_max)
                                        p ≔ a point lies on the edge {h_left.e} and at distance
                                        d
                                                away from the vertex v
                                        if the edge {h.twin_h.v, p} does not intersect with
                                        other edges improperly then
                                                left-split the vertex v
                                                counter ≔ counter + 1
                        /* right-split */
                        if dir == "right" then
                                h_right ≔ h.twin_h.prev_h.twin_h
                                if L(h_right.e) ≥ 2d_min
                                and is_right_turn (h.twin_h.v, h.v, h_right.v)
                                and dev_angle (h.twin_h.v, h.v, h_right.v) ≥ 5° then
                                        d ≔ random (d_min, d_max)
                                        p ≔ a point lies on the edge {h_right.e} and at
                                        distance d
                                                away from the vertex v
                                        if the edge {h.twin_h.v, p} does not intersect with
                                        other edges improperly then
                                                right-split the vertex v
                                                counter ≔ counter + 1
```

In the pseudocode, *random* (*a*, *b*) refers to a procedure which returns a random floating-point number N such that $a \leqslant N \leqslant b$ for $a \leqslant b$, and *random_choice* (*s*) refers to a procedure which returns a random element from the set *s*. We use *is_left_turn* (*A*, *B*, *C*) to refer to a procedure which returns true if the vertices *A*, *B*, and *C* form a path that

indicates a left-turn and returns false otherwise. Similarly, we use *is_right_turn* (*A*, *B*, *C*) to refer to a procedure which returns true if the vertices *A*, *B*, and *C* form a path that indicates a right-turn and returns false otherwise. In addition, *dev_angle* (*A*, *B*, *C*) refers to a procedure which returns the angle of deviation involved in traveling from *A* through *B* to *C*.

9.3 Quantitative Comparison

In this section, we analyze and compare six groups of designs generated by the algorithm just described. To generate the six groups of designs, we varied only one control parameter—namely, *T*, the total number of operations to be performed—and kept all the other non-random parameters the same: l_x = 800 m, l_y = 800 m, X = 9, Y = 9, τ = 4, $d_{min} = d_{max}$ = 25 m.

9.3.1 A note on determining how many times to apply the operation

To generate the six groups of designs, we first need to determine, for each group of designs, how many times to apply the operation on the initial 9 × 9 square-grid design—that is, the values of *T*. It is easy to see that based on our parameter setting, only the vertices of degree 4 (i.e., the cross intersections) in the initial street graph can be split. A vertex of degree 4 is always split into two vertices of degree 3—in other words, a cross-intersection is always split into two T-intersections. Therefore, the maximum number of times we can apply the operation of splitting vertex on the initial 9 × 9 grid is the total number of vertices that have degree 4 in the graph, which is 7 × 7 = 49. This became the largest *T* value we used to generate the superblock designs. The minimum number of times to apply the operation, T_{min}, was subsequently determined by finding the

nearest integer for 49 / 6, which is 8. Then we simply set *T* to integer multiples of 8 to generate the other groups of designs.

Examples of designs from each group are shown in Figure 186.

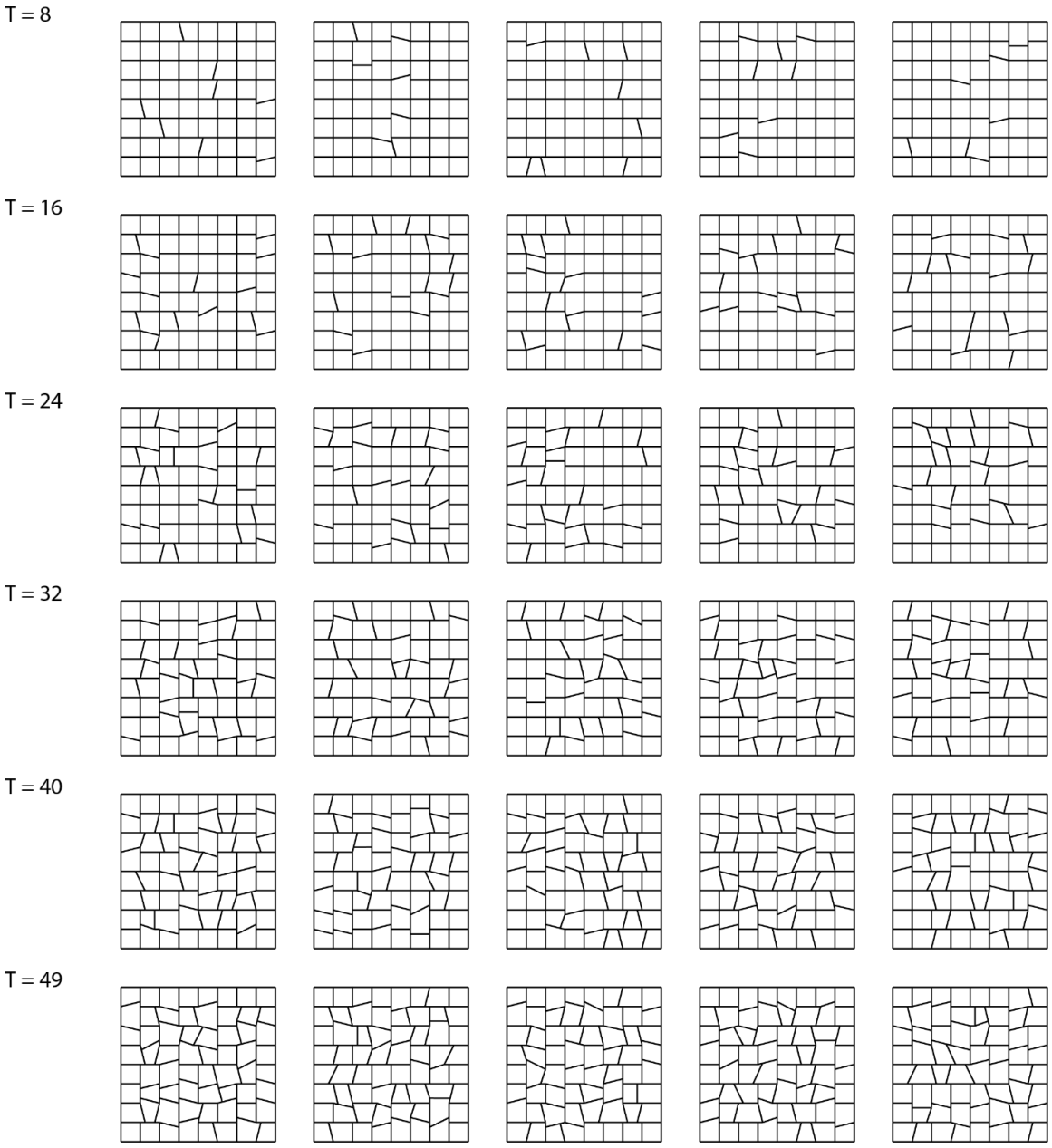

Figure 186. Examples of designs from each group (split vertex).

9.3.2 Data analysis

Six groups of designs were generated by applying the operation of splitting vertex 8, 16, 24, 32, 40, and 49 times, respectively, on the initial 9 × 9 square-grid design. Each group consists of 100 designs. The different groups of designs are analyzed and compared based on measures that characterize distinct aspects of designs.

Elementary graph properties

(1) Number of vertices

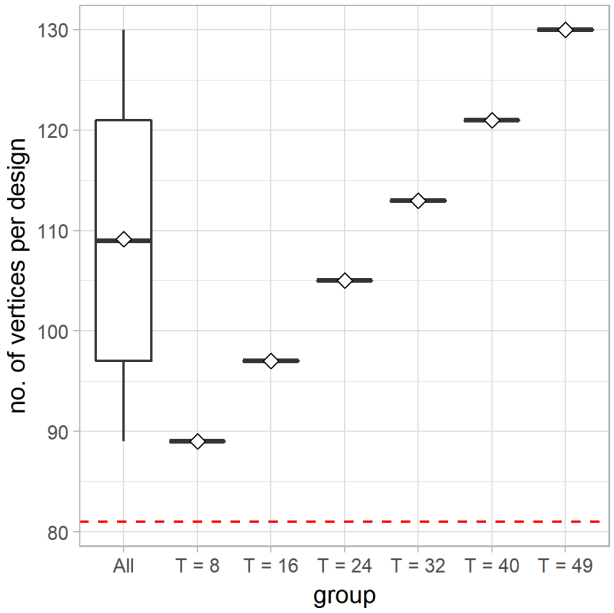

Figure 187. A boxplot (with mean diamonds) showing the distribution of the total number of vertices per design for all designs and for each group of designs (split vertex). The red dashed line indicates the total number of vertices in the initial 9 × 9 square-grid design (81).

By definition, by splitting a vertex, we create a new vertex. Therefore, as shown in Figure 187, the total number of vertices per design increases as the operation is applied

more frequently. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the total number of vertices, $|V|$, equals 81 + T.

(2) Number of edges

Every time we split a vertex, we pull them apart and create a new edge between them. Therefore, as shown in Figure 188, the total number of edges per design increases as the operation is applied more frequently. More specifically, for a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the total number of edges, $|E|$, equals 144 + T.

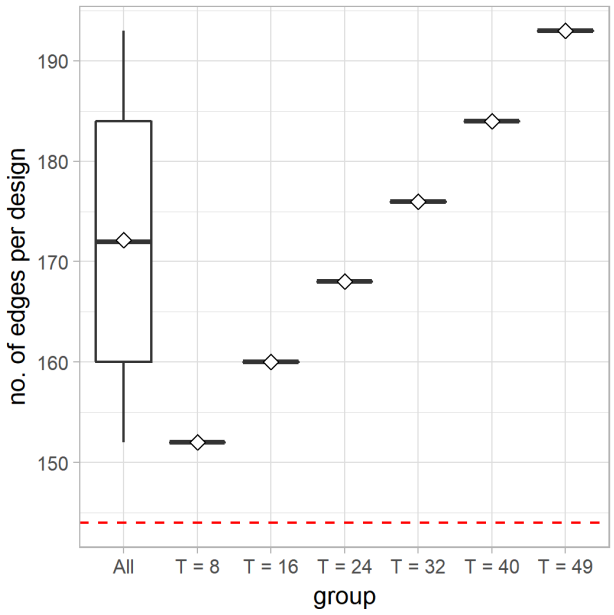

Figure 188. A boxplot (with mean diamonds) showing the distribution of the total number of edges per design for all designs and for each group of designs (split vertex). The red dashed line indicates the total number of edges in the initial 9 × 9 square-grid design (144).

(3) Number of cells

The total number of cells per design stays the same throughout the generative process.

(4) Vertex degree

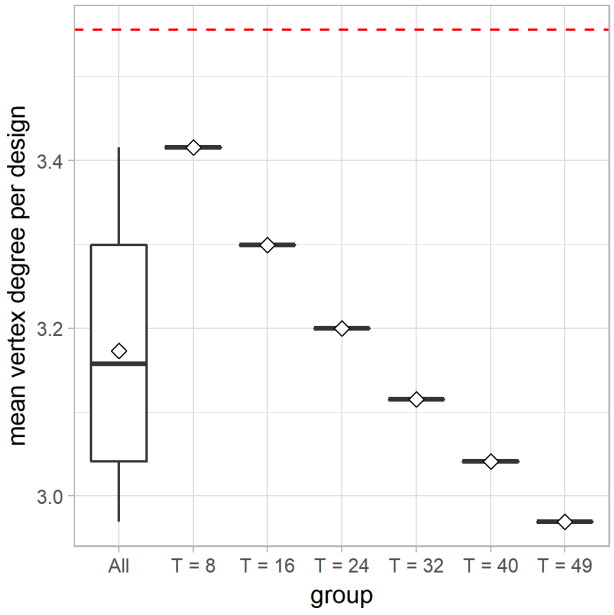

Figure 189. A boxplot (with mean diamonds) showing the distribution of the mean vertex degree per design for all designs and for each group of designs (split vertex). The red dashed line indicates the mean vertex degree for the initial 9 × 9 square-grid design (≈ 3.556)

Each application of the operation of splitting vertex decreases the degree of the original vertex by one and creates a new vertex of degree 3. Based on our algorithm and the parameter setting, each time we apply the operation based on the initial square-grid design, we convert a vertex of degree 4 into two vertices of degree 3. Therefore, as shown in Figure 189, when analyzed in groups, the mean vertex degree per design decreases steadily as T increases. Based on the handshaking theorem, the sum of the

degrees of the vertices of a graph is twice the number of its edges. Therefore, the mean vertex degree for a graph is $2|E| / |V|$. For a design generated by applying the operation T times on the initial 9 × 9 square-grid design, the mean vertex degree equals $2(144 + T) / (81 + T)$.

Density of streets, blocks, intersections, and connectivity

(1) Total street length per design

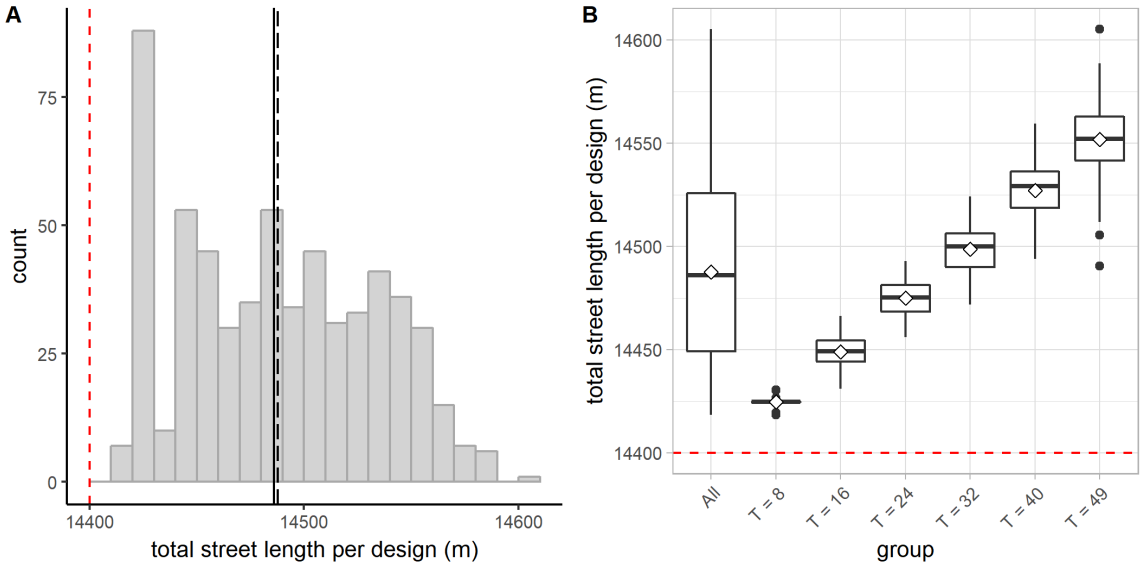

Figure 190. Distribution of the total street length per design (split vertex). A: A histogram showing the distribution of the total street length per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the total street length for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the total street length per design for all designs and for each group of designs.

For all the designs analyzed, the total street length per design ranges from 14418 to 14605 m. In all cases, the total street length per design is greater than 14400 m—the total street length of the initial 9 × 9 square-grid design. This finding is not surprising

because as a vertex is split in the initial square grid—be it a left-split or a right-split—the edge being swung by the split-operation is often stretched to become longer. As shown in Figure 190B, when analyzed in groups, as *T* increases, the total street length per design tends to increase as well.

(2) Total number of blocks per design

The operation of splitting vertex never subdivides or merges blocks. Therefore, the total number of blocks per design stays the same throughout the generative process.

(3) Total number of intersections per design

Based on our algorithm and the parameter setting, each time we split a vertex, we convert a vertex of degree 4 into a vertex of degree 3 and also create another vertex of degree 3. Therefore, as more vertices are split, the total number of T-intersections increases at a faster pace than the loss of cross-intersections. As shown in Figure 191, when analyzed in groups, the total number of intersections per design increases as *T* increases. More specifically, the total number of intersections in a design generated by applying the operation of splitting vertex *T* times on the initial 9 × 9 square-grid design equals 77 + *T*.

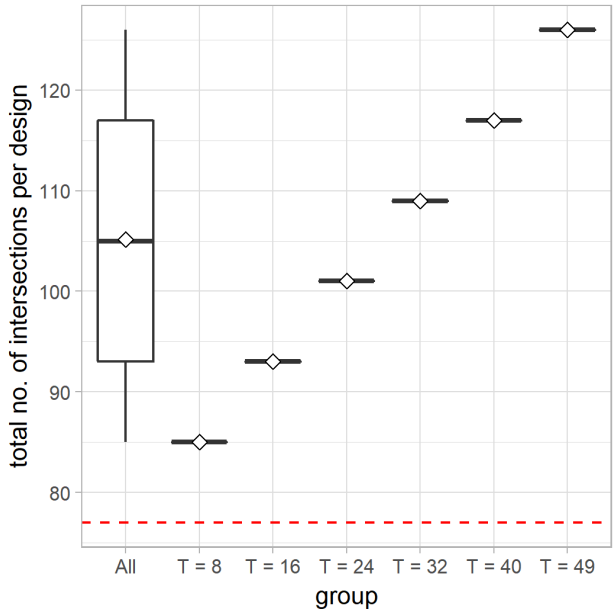

Figure 191. A boxplot showing the distribution of the total number of intersections per design for all designs and for each group of designs (split vertex). The red dashed line indicates the total number of intersections in the initial 9 × 9 square-grid design (77).

(4) Distance between intersections

By definition, after a vertex is split, the vertices are pulled apart by swinging one of the edges that are incident to the original vertex along another incident edge. In doing so, the edge being swung along is also split into two by the new vertex placed on it. Thus, the operation can create dramatically shorter distances between intersections. (Figure 192B). Although during the vertex-split, the edge being swung is often stretched a little, the slight increase in the edge length cannot counteract the effect of the dramatic reduction of distance between intersections just mentioned. Therefore, as shown in Figure 192C, when analyzed in groups, the mean distance between intersections per design decreases as *T* increases.

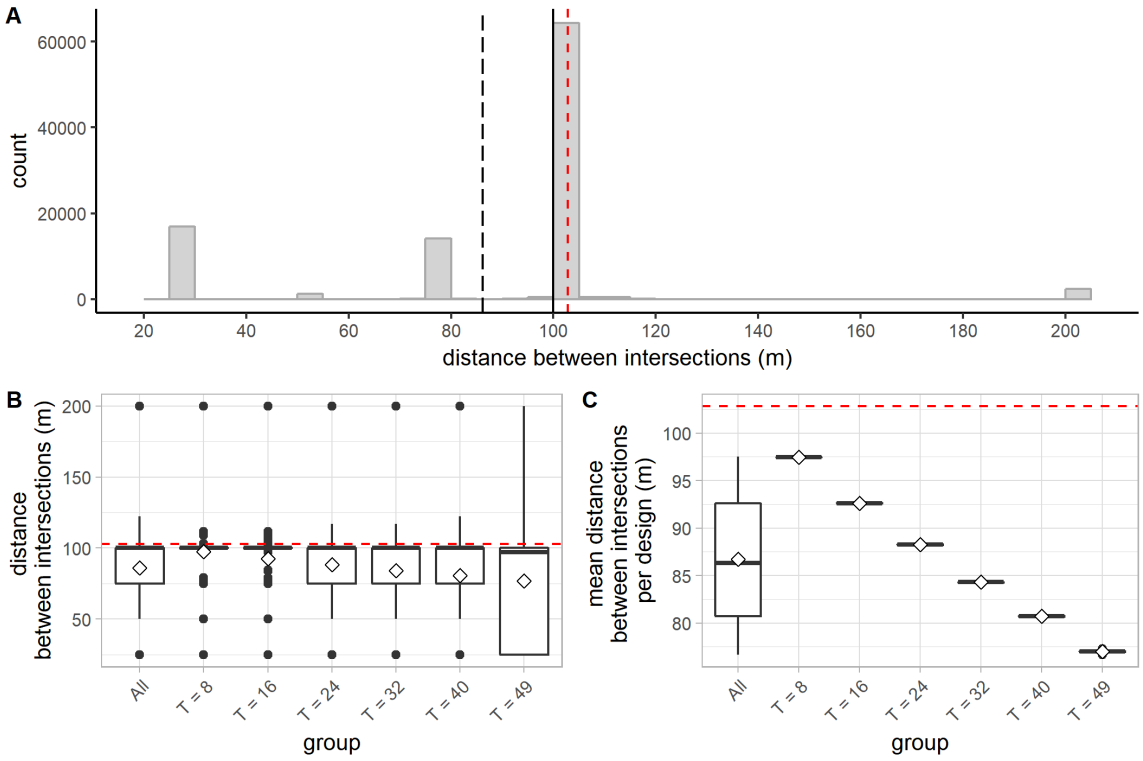

Figure 192. Distribution of the distance between intersections (split vertex). A: A histogram showing the distribution of the distance between intersections for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean distance between intersections for the initial 9 × 9 square-grid design (≈ 102.86 m). B: A boxplot showing the distribution of the distance between intersections observed in all designs and in each group of designs. C: A boxplot showing the distribution of the mean distance between intersections per design for all designs and for each group of designs.

Directional reach and directional distance

(1) DDL

Based on our algorithm and the setup of the initial street graph, the angles of deviation are either minuscule (< 15°) or very large. Therefore, while the distribution of DDL10d looks slightly different, the distributions of DDL20d and DDL30d look identical for all the segments analyzed (Figure 193). To be consistent with other chapters, in the

following analysis, unless otherwise specified, the directional distance is always evaluated by setting the threshold angle to 20°.

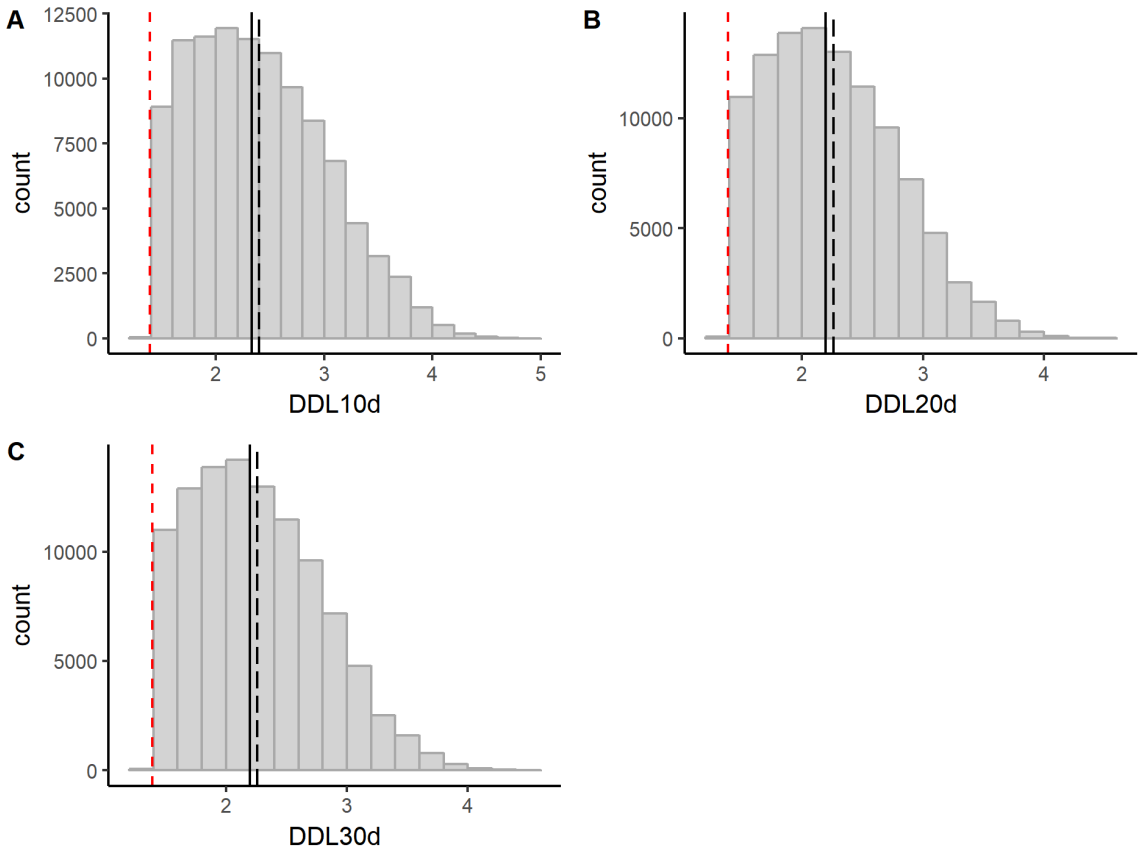

Figure 193. Distributions of DDL10d, DDL20d, and DDL30d (split vertex). In each subfigure, the long-dashed line indicates the mean; the solid line indicates the median; the red dashed line indicates the corresponding mean DDL value for the initial 9 × 9 square-grid design.

The DDL20d values for all the segments analyzed range from 1.39 to 4.56. As shown in Figure 193B, more than 60% of all segments assume a DDL20d value greater than 2. Based on our algorithm and the parameter setting, the operation of splitting vertex tends to increase the mean directional distance of a street system because it breaks the linear flow of streets which form the cross-intersections. As shown in Figure 194, more frequent application of this operation tends to produce more segregated streets. When

analyzed in groups, the mean DDL20d per design clearly tends to increase as *T* increases (Figure 194B).

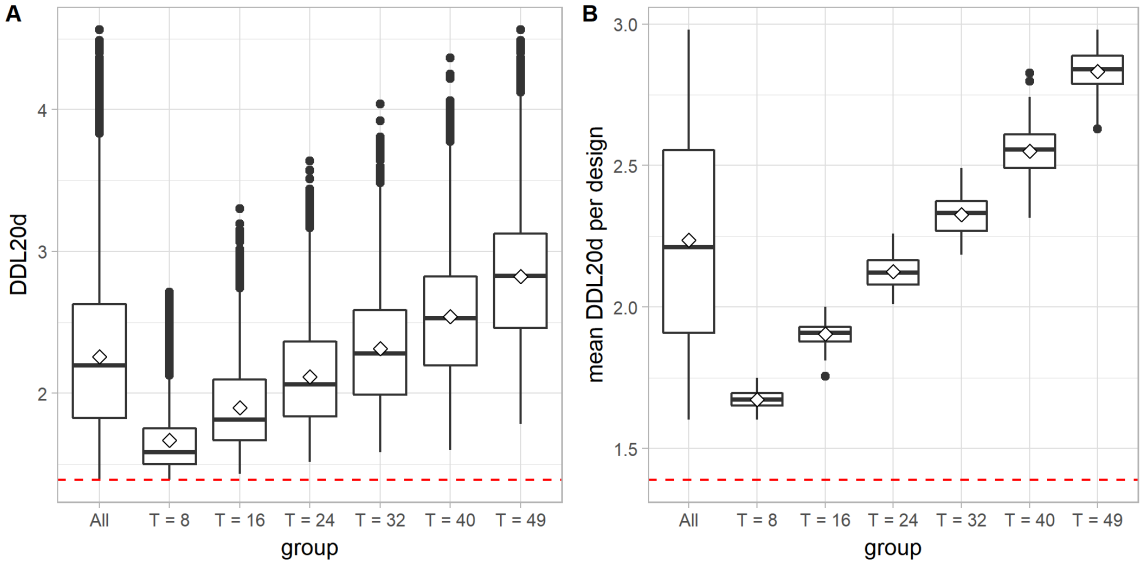

Figure 194. Distribution of DDL20d (split vertex). A: A boxplot (with mean diamonds) showing the distribution of DDL20d values for all segments and for segments in each group of designs. In each subfigure, the red dashed line indicates the mean DDL20d for the initial 9 × 9 square-grid design (≈ 1.389). B: A boxplot (with mean diamonds) showing the distribution of the mean DDL20d per design for all designs and for each group of designs.

(2) Linear reach (dr0dc20d)

As shown in Figure 195A, the linear reach values are clustered around integer multiples of the typical distance between intersections (100 m) in the initial square grid. It is not surprising because by splitting vertices on the horizontal or vertical traversal street in the initial square-grid design, one can generate segments that have linear extensions spanning either few or many street intervals. The operation tends to reduce the linear reach as it breaks the linear flow of streets. As shown in Figure 195C, when analyzed in groups, the mean linear reach per design decreases as *T* increases.

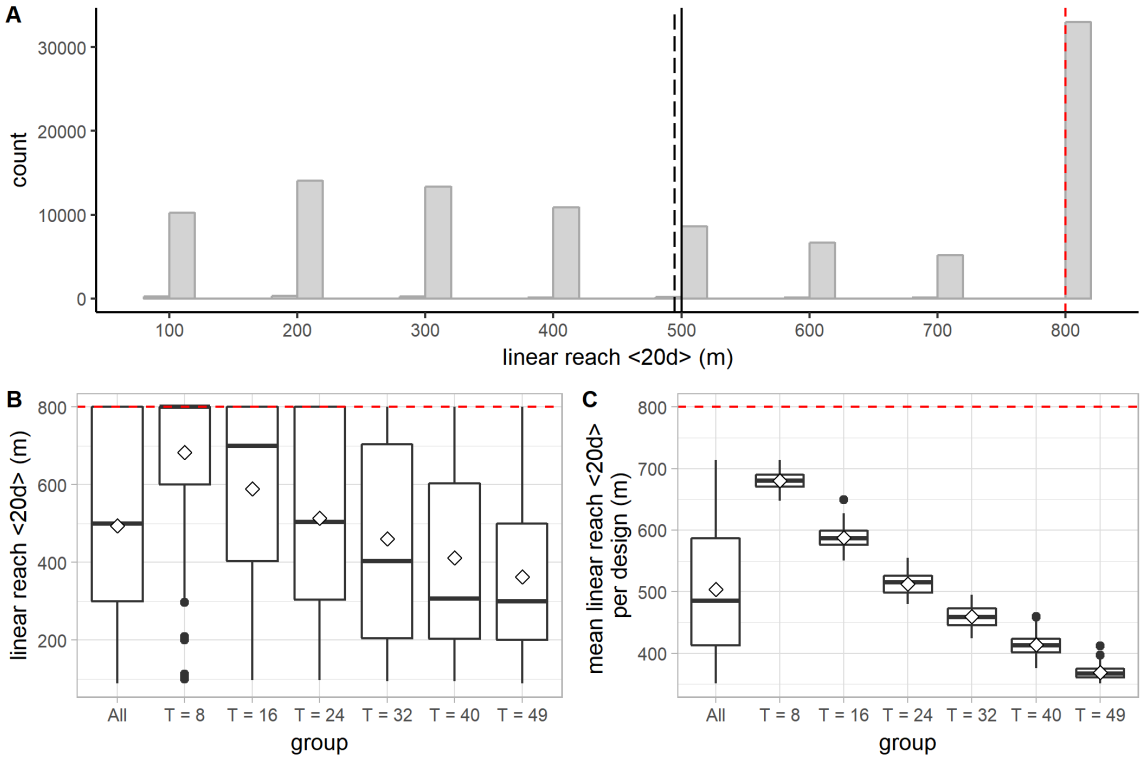

Figure 195. Distribution of the linear reach (split vertex). A: A histogram showing the distribution of the linear reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean linear reach for the initial 9 × 9 square-grid design (800 m). B: A boxplot (with mean diamonds) showing the distribution of the linear reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean linear reach per design for all designs and for each group of designs.

(3) 2-dc reach (dr2dc20d)

The 2-dc reach values assumed by all the segments analyzed range from around 1300 m to about 14500 m—which is a fairly wide range. The 2-dc reach for a segment tends to drop in designs generated by more frequent split of vertices (Figure 196B). While the total street length per design tends to increase as more vertices are split, the mean 2-dc reach per design tends to decrease (Figure 196C). Such kind of trend is

different from, say, the operation of linking edge to edge and the operation of linking vertex to edge.

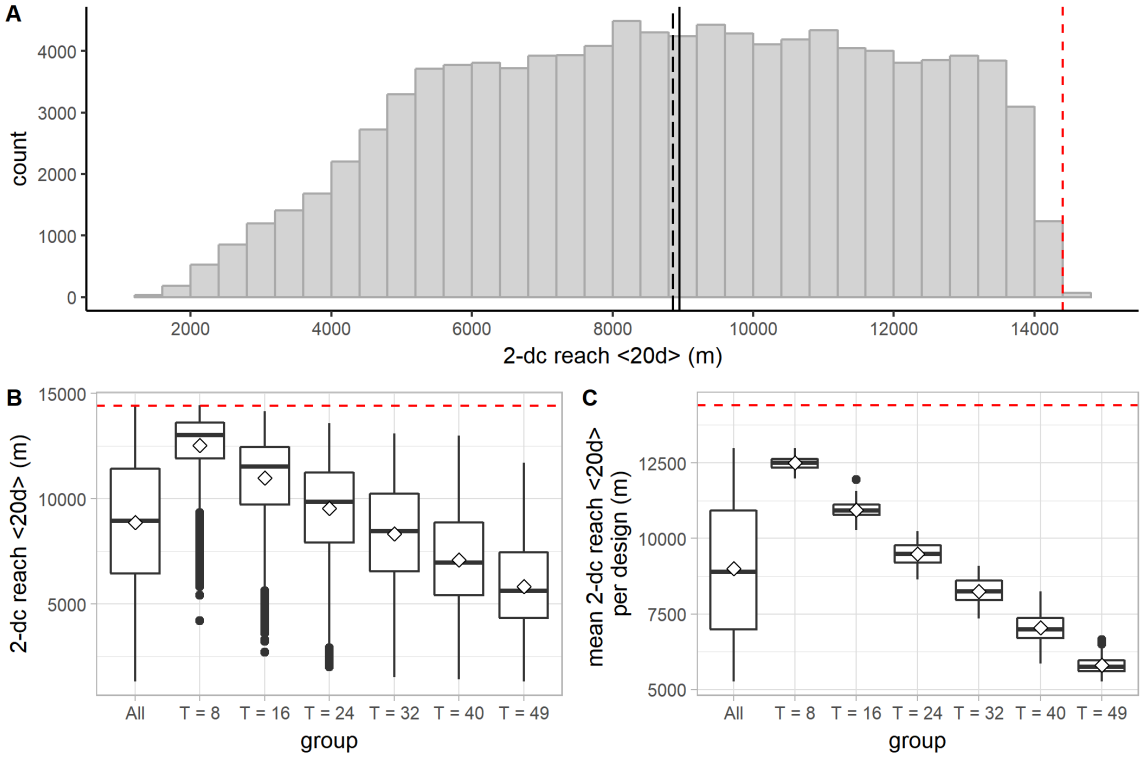

Figure 196. Distribution of the 2-dc reach (split vertex). A: A histogram showing the distribution of the 2-dc reach for all segments in all groups of designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean 2-dc reach for the initial 9 × 9 square-grid design (14400 m). B: A boxplot (with mean diamonds) showing the distribution of the 2-dc reach for all segments and for segments in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean 2-dc reach per design for all designs and for each group of designs.

Geometric regularity

(1) Fragmentality per design

For all the designs analyzed, the fragmentality per design ranges from 0.17 to 0.36. Since the operation of splitting vertex tends to break the linear flow of streets, it

increases the total number of continuity lines formed in the system. As a result, the design becomes more fragmented in that the ratio between the total number of continuity lines and the total number of edges increases. As shown in Figure 197B, when analyzed in groups, the fragmentality per design tends to increase as *T* increases.

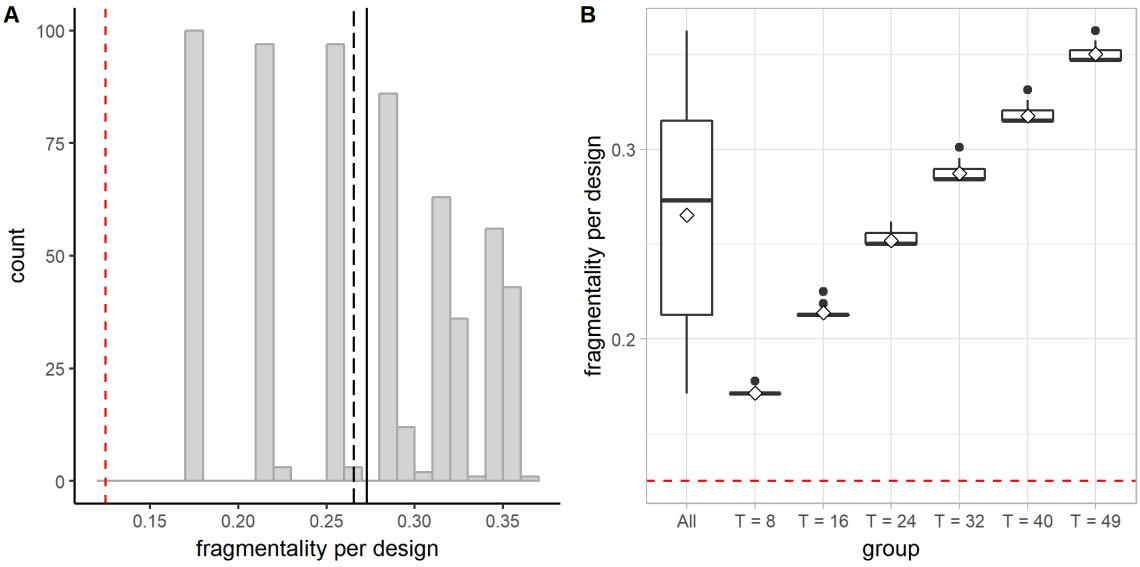

Figure 197. Distribution of the fragmentality per design (split vertex). A: A histogram showing the distribution of the fragmentality per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the fragmentality for the initial 9 × 9 square-grid design (0.125). B: A boxplot (with mean diamonds) showing the distribution of the fragmentality per design for all designs and for each group of designs.

(2) Block area

For all the blocks analyzed, the block area ranges from 5000 to 15000 m². As shown in Figure 198, the distribution looks symmetric, partially because the operation simultaneously changes the sizes of two adjacent blocks, making one of them bigger and the other smaller, by the same amount. Since the total number of blocks stays the same

throughout the generative process, the mean block area per design stays the same no matter how many times the operation is applied.

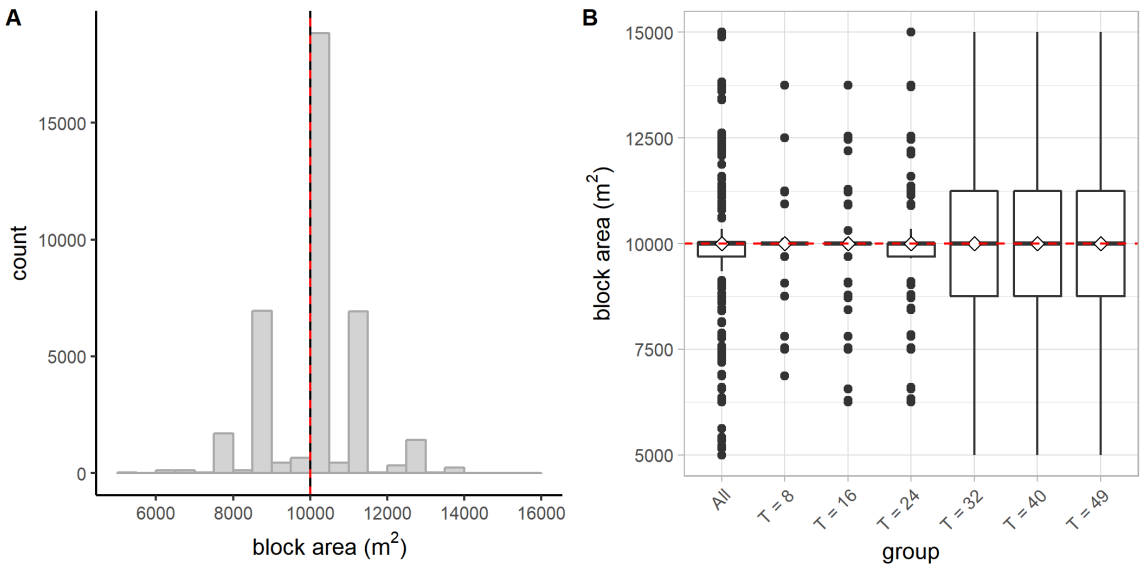

Figure 198. Distribution of the block area (split vertex). A: A histogram showing the distribution of the area for all the blocks in all designs. The solid black line indicates the median. In both subfigures, the red dashed line indicates the mean block area for the initial 9 × 9 square-grid design (10000 m²). B: A boxplot showing the distribution of the area for all blocks and for blocks in each group of designs.

While the mean block area per design stays the same throughout the generative process, the areas of the blocks become more and more varied as the operation is applied more frequently, as evidenced by the increasing standard deviation of the block area per design (Figure 199).

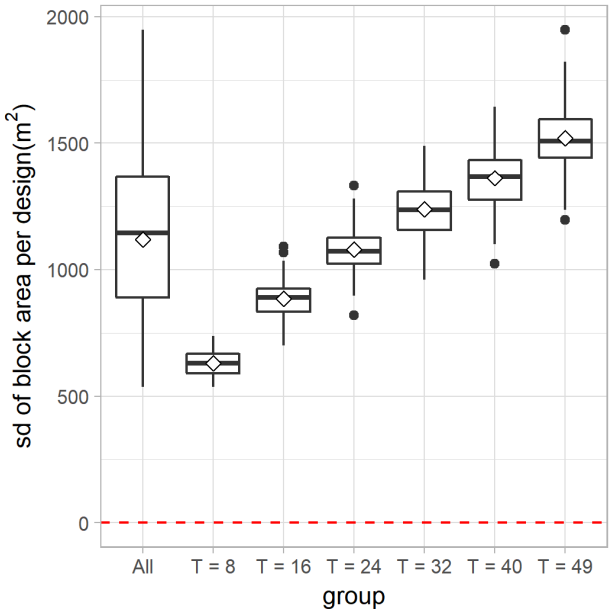

Figure 199. A boxplot showing the distribution of the standard deviation of the block area per design for all designs and for each group of designs (split vertex). The red dashed line indicates the standard deviation of the block area for the initial 9 × 9 square-grid design (0 m²).

(3) Block perimeter

For all the blocks analyzed, the block perimeter ranges from 294 to 512 m. As shown in Figure 200A, the distribution looks symmetric, partially because the operation always simultaneously affects two adjacent cells. Since the total number of blocks per design remains the same, the mean block perimeter per design is a direct function of the total street length per design. As shown in Figure 200C, when analyzed in groups, the mean block perimeter per design increases as *T* increases.

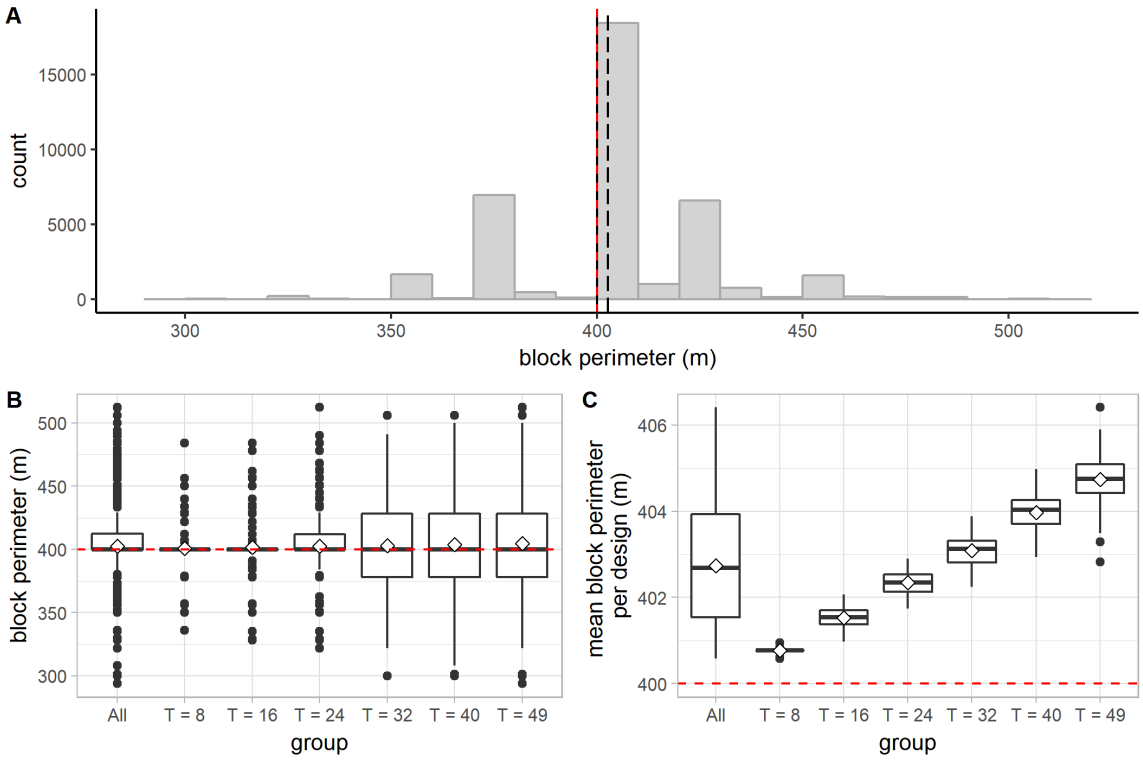

Figure 200. Distribution of the block perimeter (split vertex). A: A histogram showing the distribution of the perimeter for all blocks in all designs. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean block perimeter for the initial 9 × 9 square-grid design (400 m). B: A boxplot (with mean diamonds) showing the distribution of the perimeter for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean block perimeter for all designs and for each group of designs.

The perimeters of blocks inside a design become more and more varied as more vertices have been split, as evidenced by the increasing standard deviation of the block perimeter per design and the increasing coefficient of variation of the block perimeter per design (Figure 201).

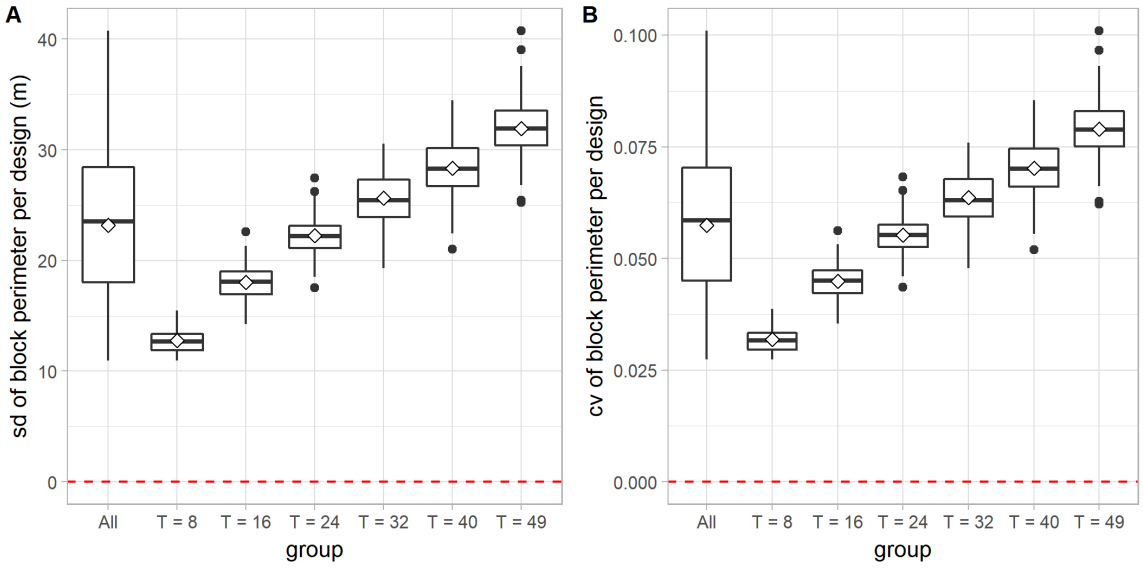

Figure 201. Variation of the block perimeter per design (split vertex). A: A boxplot showing the distribution of the standard deviation of the block perimeter per design for all designs and for each group of designs. The red dashed line indicates the standard deviation of the block perimeter for the initial 9 × 9 square-grid design (0 m). B: A boxplot showing the distribution of the coefficient of variation (CV) of the block perimeter per design for all designs and for each group of designs. The red dashed line indicates the CV of the block perimeter per design for the initial 9 × 9 square-grid design (0.0).

(4) Standardized block area-perimeter ratio (SAPR)

For all the blocks analyzed, the standardized block area-perimeter ratio (SAPR) ranges from 0.92 to 1.01—a relatively narrow range. As shown in Figure 202B, the blocks produced by more frequent application of the operation of splitting vertex tend to be more varied in SAPR, as evidenced by the increasingly wider range of values. When analyzed in groups, the mean SPAR per design tends to decrease as *T* increases (Figure 202C).

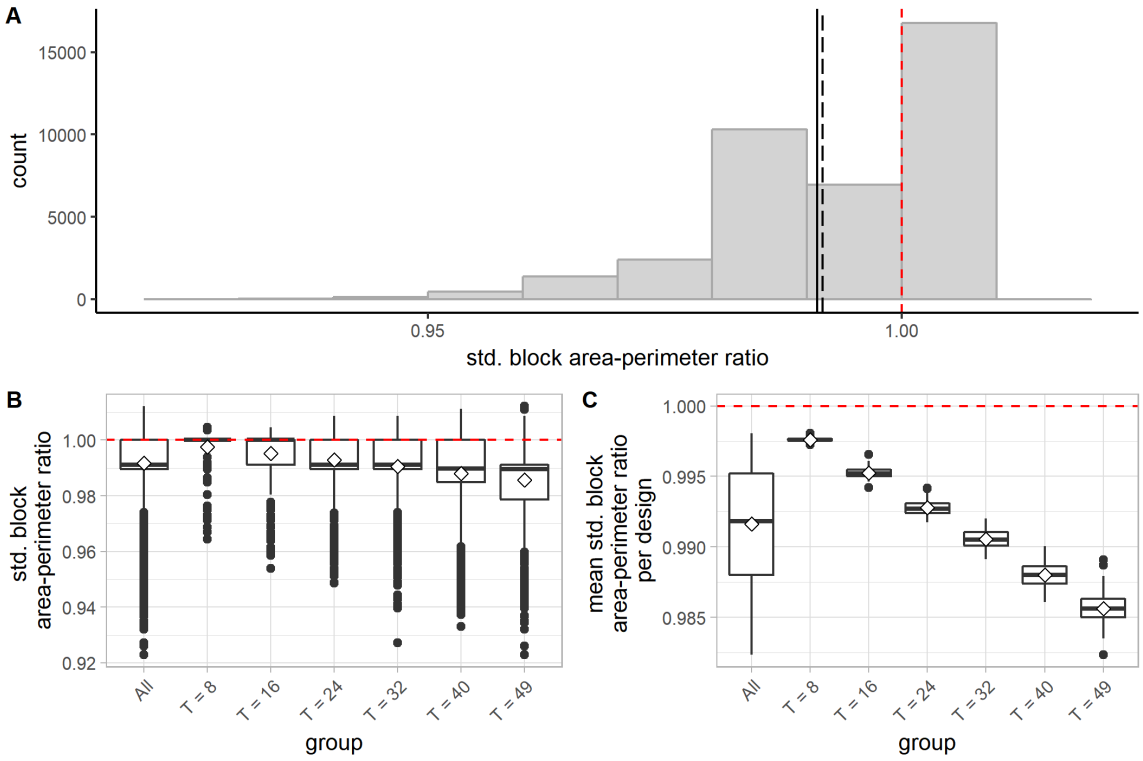

Figure 202. Distribution of the standardized block area-perimeter ratio (SAPR) (split vertex). A: A histogram showing the distribution of SAPR for all blocks. The long-dashed line indicates the mean; the solid line indicates the median. In all the subfigures, the red dashed line indicates the mean SAPR for the initial 9 × 9 square-grid design (1.00). B: A boxplot (with mean diamonds) showing the distribution of SAPR for all blocks and for blocks in each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the mean SAPR per design for all designs and for each group of designs.

Diversity in syntactic conditions

(1) Total number of and proportion of distinct DDL20d values per design

As shown in Figure 203B, when analyzed in groups, the total number of distinct DDL20d values per design tends to increase as *T* increases. Noticing that the total number of segments/edges also increases during the generative process, we also plotted the proportion of distinct DDL20d values per design to counteract the effect of the

growing size of the street graph. For all the designs analyzed, the proportion of distinct DDL20d values per design ranges from 0.125 to 0.347. Similarly, when analyzed in groups, the proportion of distinct DDL20d values per design increases as *T* increases (Figure 203C).

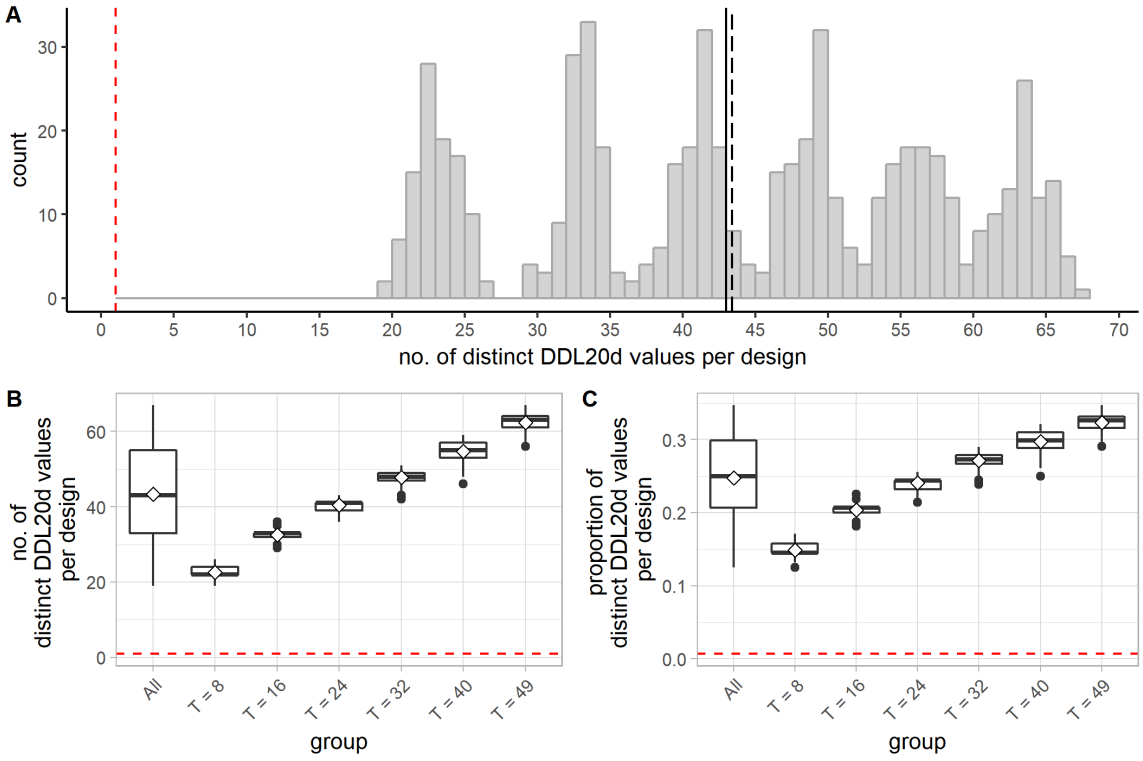

Figure 203. Distribution of the total number and the proportion of distinct DDL20d values per design (split vertex). A: A histogram showing the distribution of the total number of distinct DDL20d values per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In this subfigure and the following one, the red dashed line indicates the total number of distinct DDL20d values for the initial 9 × 9 square-grid design (1). B: A boxplot (with mean diamonds) showing the distribution of the total number of distinct DDL20d values per design for all designs and for each group of designs. C: A boxplot (with mean diamonds) showing the distribution of the proportion of distinct DDL20d values per design for all designs and for each group of designs. The red dashed line indicates the proportion of distinct DDL20d values for the initial 9 × 9 square-grid design (≈ 0.007).

(2) Standard deviation of DDL20d values per design

For all the designs analyzed, the standard deviation of DDL20d per design ranges from 0.21 to 0.54. As shown in Figure 204B, when analyzed in groups, the standard deviation of DDL20d per design tends to increase as *T* increases.

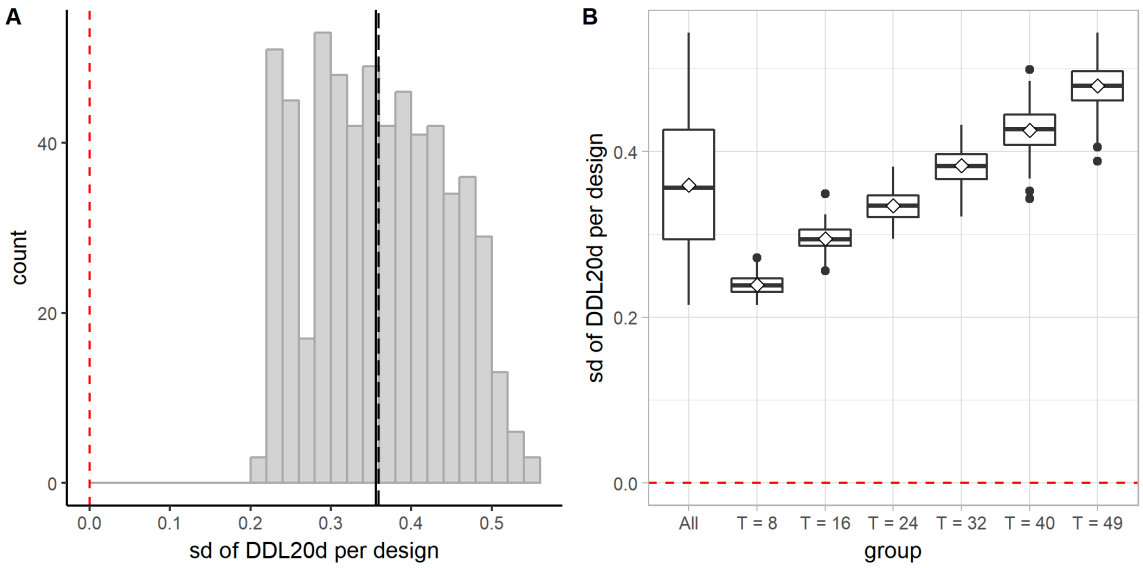

Figure 204. Distribution of the standard deviation of DDL20d per design (split vertex). A: A histogram showing the distribution of the standard deviation of DDL20d per design for all designs. The long-dashed line indicates the mean; the solid line indicates the median. In each subfigure, the red dashed line indicates the standard deviation of DDL20d for the initial 9 × 9 square-grid design (0.0). B: A boxplot (with mean diamonds) showing the distribution of the standard deviation of DDL20d per design for all designs and for each group of designs.

9.4 Discussion and Conclusion

The impacts are discussed based on the assumption that the initial street graph resembles a square grid.

The impact of the operation on the graph properties: Each application of the operation increases both the total number of vertices and the total number of edges by one. The total number of cells stays the same throughout the generative process. Each application of the operation decreases the degree of the original vertex by one and creates a new vertex of degree 3. Thus, the operation decreases the mean vertex degree per design.

The impact of the operation on the density of a street network: The operation tends to increase the total street length per design. The number of blocks per design, however, stays the same no matter how many times the operation is applied. By creating an additional T-intersection, each application of the operation increases the total number of intersections by one. The operation also tends to decrease the mean distance between intersections per design because there are more intersections interspersed in the street network, with the total street length increased only slightly.

The impact of the operation on the directional reach/distance: By disrupting the linear flow of streets, the operation tends to increase the mean DDL20d per design. For the same reason, it tends to reduce the mean linear reach and the mean 2-dc reach per design.

The impact of the operation on the regularity of a street network: The operation tends to increase the fragmentality per design. As the operation is applied more frequently, the area and perimeter of the blocks in a design also become more varied. The mean standard block area-perimeter ratio per design tends to drop as more vertices are split.

The impact of the operation on the diversity of syntactic conditions: The operation tends to increase the diversity of syntactic conditions in that both the total number of DDL20d values and the proportion of DDL20d values per design tend to increase as more operations are applied. The syntactic conditions within a design also tend to be more varied as the operation is applied more frequently.

CHAPTER 10
COMPARISON OF GENERATIVE RULES

In previous chapters, we have introduced eight different types of operations to deform a street graph. For each type of operation, we have developed a set of rules (presented as algorithms) to generate superblock designs. While we have compared designs generated by the same set of rules, we have not compared designs generated by different rulesets. Since only one set of rules is developed for each type of operation, for convenience purposes, we would refer to each ruleset by the type of operation it involves. In this chapter, we group the designs by the type of operation involved in the generative process, and we compare the effects caused by the different types of operations—or more accurately, the effects caused by the different generative rules that are associated with the different types of operations.

To make a comprehensive comparison, we have also generated an additional set of designs by randomly applying the operations of disjoining vertices, linking edge to edge, shifting vertex, splitting vertex, and cross-concatenating vertices a specified number of times. We refer to this generative ruleset as “mixed operations”. Examples of designs generated by the mixed operations are shown in Figure 205. A total of six hundred designs were generated by the mixed operations, with T (i.e., the total number of operations to apply) set to 10, 20, 30, 40, 50, and 60, respectively. Therefore, with this additional set of designs, we compare a total of 5400 designs generated by 9 different types of operations.

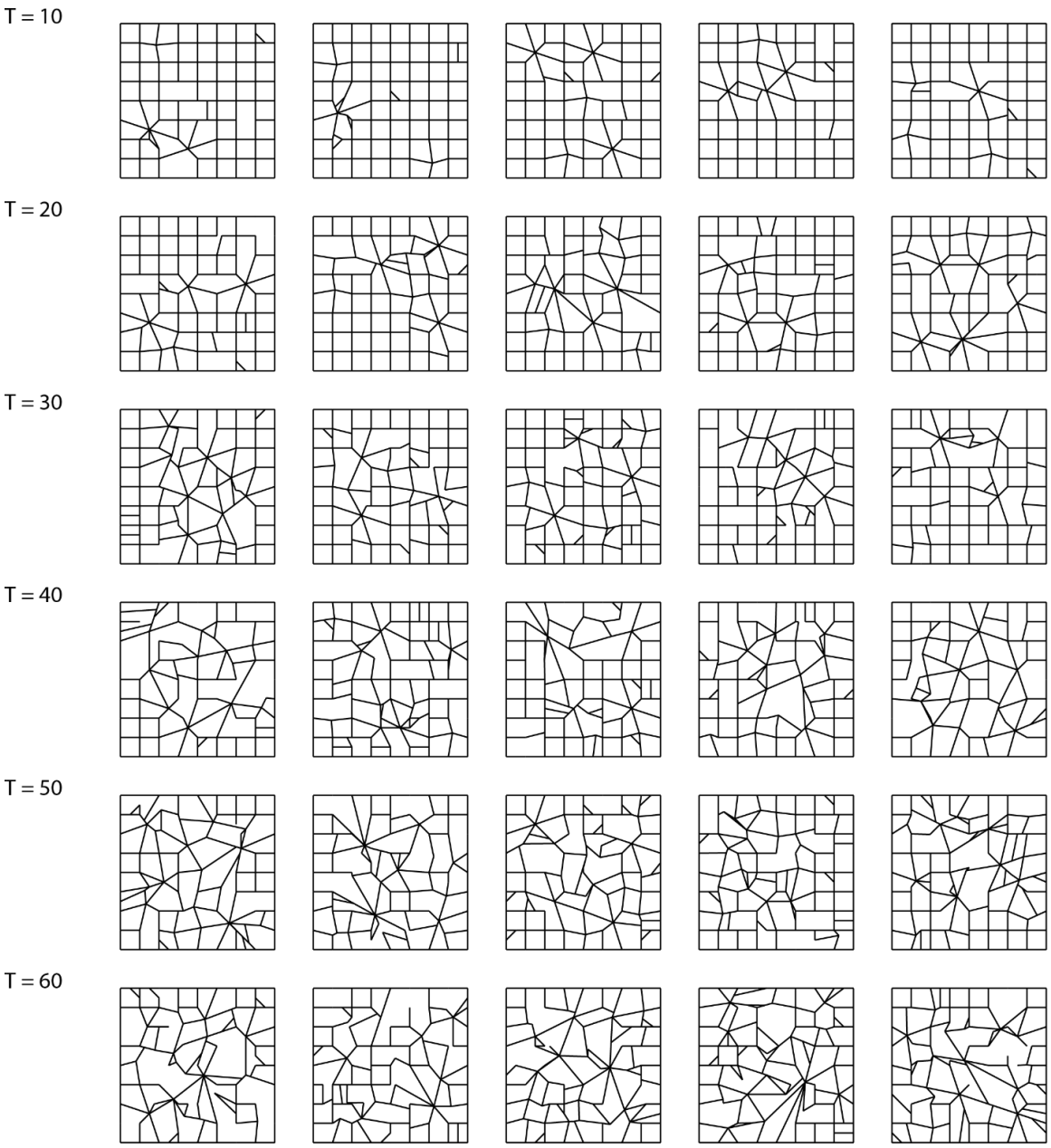

Figure 205. Examples of designs generated by the mixed operations.

In the first half of the chapter, we compare the group means across different measures. This gives us a sense of the operations' and the generative rules' different capacities to influence individual measures. In the second half of the chapter, we quantify the relationship between measures with simple linear regression models. By finding a

consistent, or inconsistent, relationship between measures for different groups of designs, we build our understanding of how different types of operations influence the resulting design typologies.

10.1 Comparison Based on Individual Measures

10.1.1 Elementary graph properties

Unlike in the previous chapters that reviewed the effects of individual rules, this chapter compares the graph properties across the different groups based upon the final polished street graphs. The reader must remember that the analytical descriptions of designs that are based on polished graphs cannot always be anticipated based on the analysis of intermediate street graphs.

(1) Total number of vertices per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the total number of vertices per design are significantly different from each other except between shifting vertex and linking vertex to vertex (Figure 206). As shown in Figure 210A, applying the operation of linking edge to edge can dramatically increase the total number of vertices in a street graph. It is not surprising because each time we link an edge to another edge, we are bound to produce two additional vertices in the street graph.

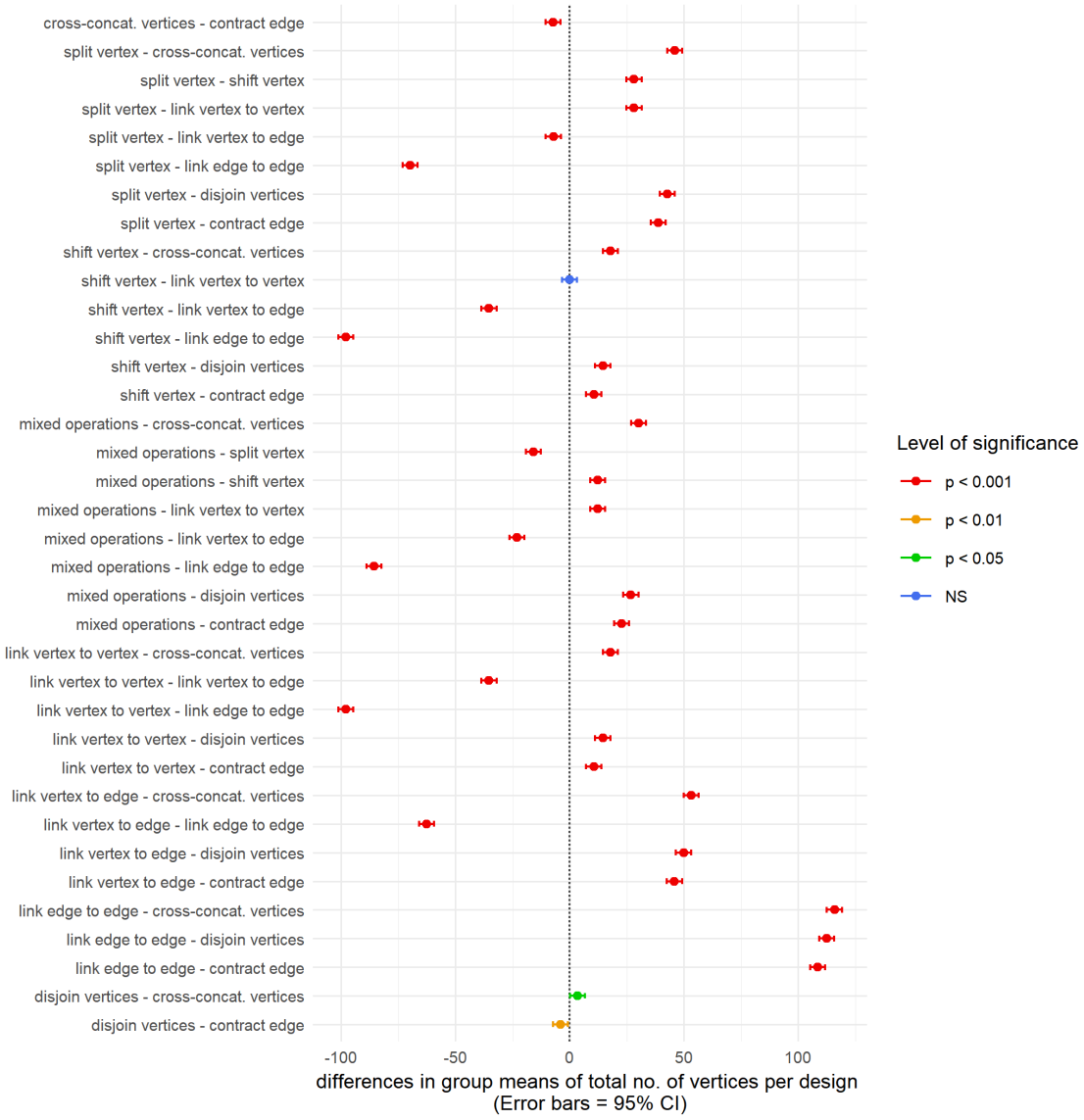

Figure 206. Differences between the group means of the total number of vertices per design and the 95% confidence intervals (Tukey's HSD test).

By contrast, linking vertex to edge and splitting vertex only add one vertex at a time. Both the operations of contracting edge and cross-concatenating vertices reduce the number of vertices. Disjoining vertices tends to reduce the total number of vertices too—note that although disjoining vertices would not change the number of vertices before the street graphs are polished, the number of vertices could be reduced during the

polishing process (for example, the redundant vertices on a long straight cul-de-sac would be removed). By definition, shifting vertex and linking vertex to vertex never changes the number of vertices in a design. In addition, the designs generated by the mixed operations have a significantly higher number of vertices per design than the initial design (One-sample Student's *t*-test, $p < .001$).

(2) Total number of edges per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the total number of edges per design are significantly different from each other (Figure 207). As shown in Figure 210B, applying the operation of linking edge to edge can dramatically increase the total number of edges in a street graph. It is not surprising because each time we link an edge to another edge, we are bound to increase the total number of edges by three. Linking vertex to edge also increases the number of edges quickly, by two at a time. Both linking vertex to vertex and splitting vertex increases the number of edges by one at a time. By definition, contracting edge and disjoining vertices reduce the number of edges by one at a time, although disjoining vertices can lose more edges because of the polishing process. Cross-concatenating vertices also decreases the number of edges, by two at a time, or even more after the polishing process. Shifting vertex never changes the number of edges. In addition, the designs generated by the mixed operations have a significantly higher number of edges per design than the initial design (One-sample Student's *t*-test, $p < .001$).

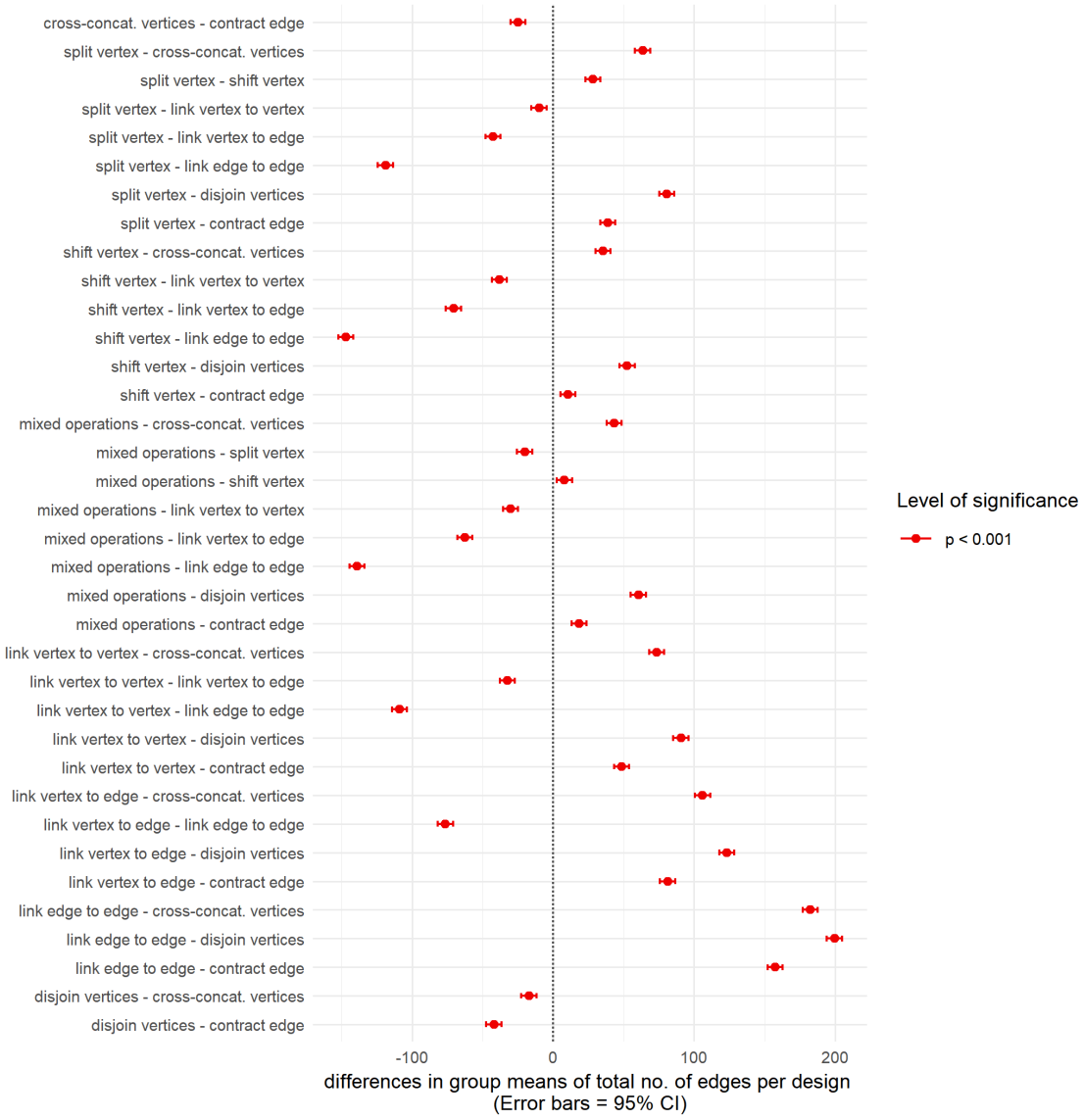

Figure 207. Differences between the group means of the total number of edges per design and the 95% confidence intervals (Tukey's HSD test).

(3) Total number of cells per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the total number of cells per design are significantly different from each other except between

splitting vertex and shifting vertex, between splitting vertex and contracting edges, and between shifting vertex and contracting edge (Figure 208).

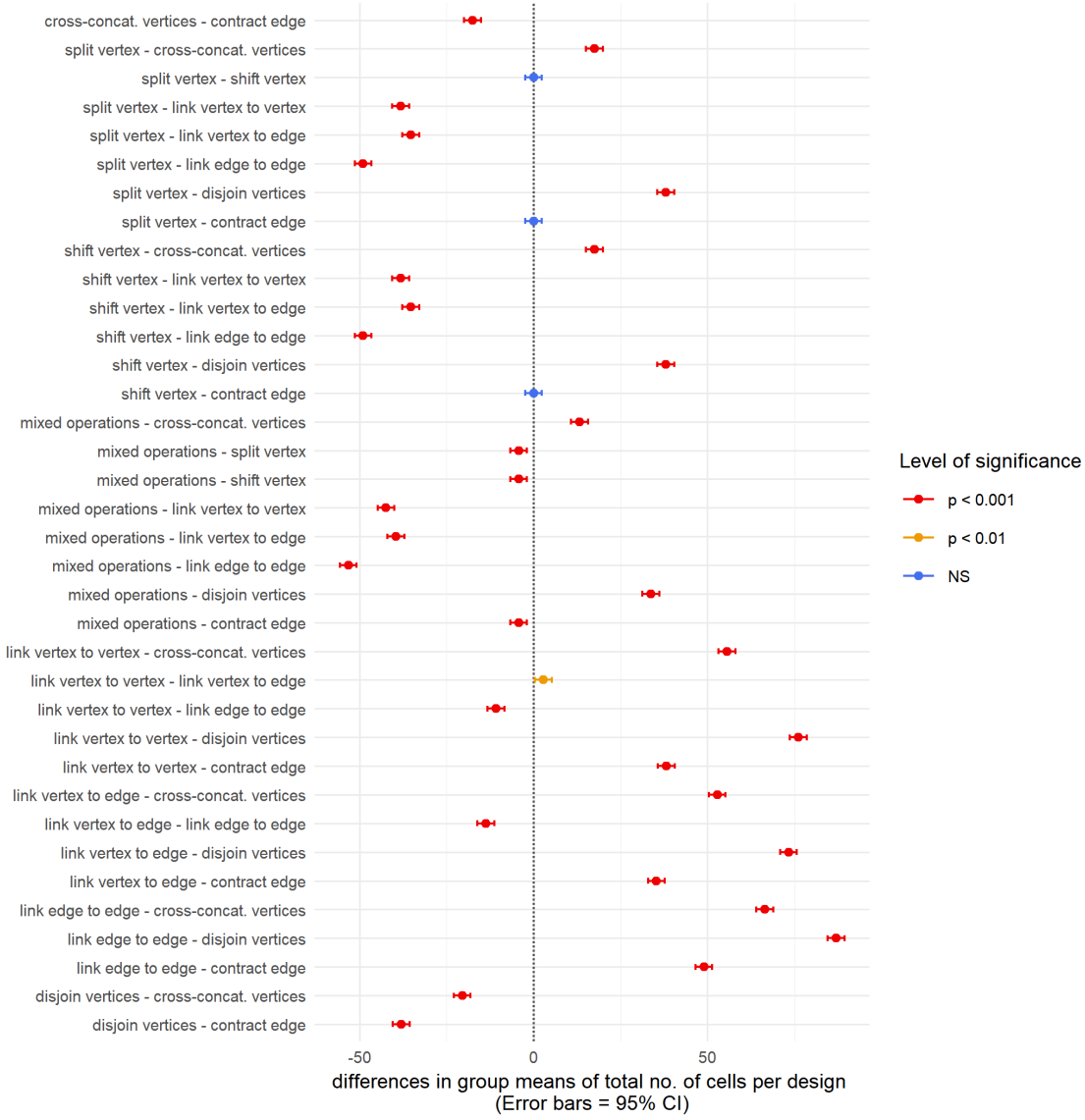

Figure 208. Differences between the group means of the total number of cells per design and the 95% confidence intervals (Tukey's HSD test).

As shown in Figure 210C, linking edge to edge, linking vertex to edge, and linking vertex to vertex all increase the number of cells/blocks in the design. By

definition, all three operations increase the number of blocks by one at a time. By contrast, cross-concatenating vertices and disjoining vertices reduce the number of cells/blocks by one at a time. Shifting vertex, splitting vertex, and contracting edge do not change the total number of cells/blocks in a design. In addition, the designs generated by the mixed operations have a significantly lower number of cells/blocks per design than the initial design (One-sample Student's *t*-test, $p < .001$).

(4) Mean vertex degree per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the mean vertex degree per design are significantly different from each other except between linking edge to edge and the mixed operations (Figure 209). As shown in Figure 210D, linking vertex to vertex can dramatically increase the mean vertex degree per design, while disjoining vertices can dramatically reduce it. The designs generated by the mixed operations have significantly lower mean vertex degree per design than the initial design (One-sample Student's *t*-test, $p < .001$).

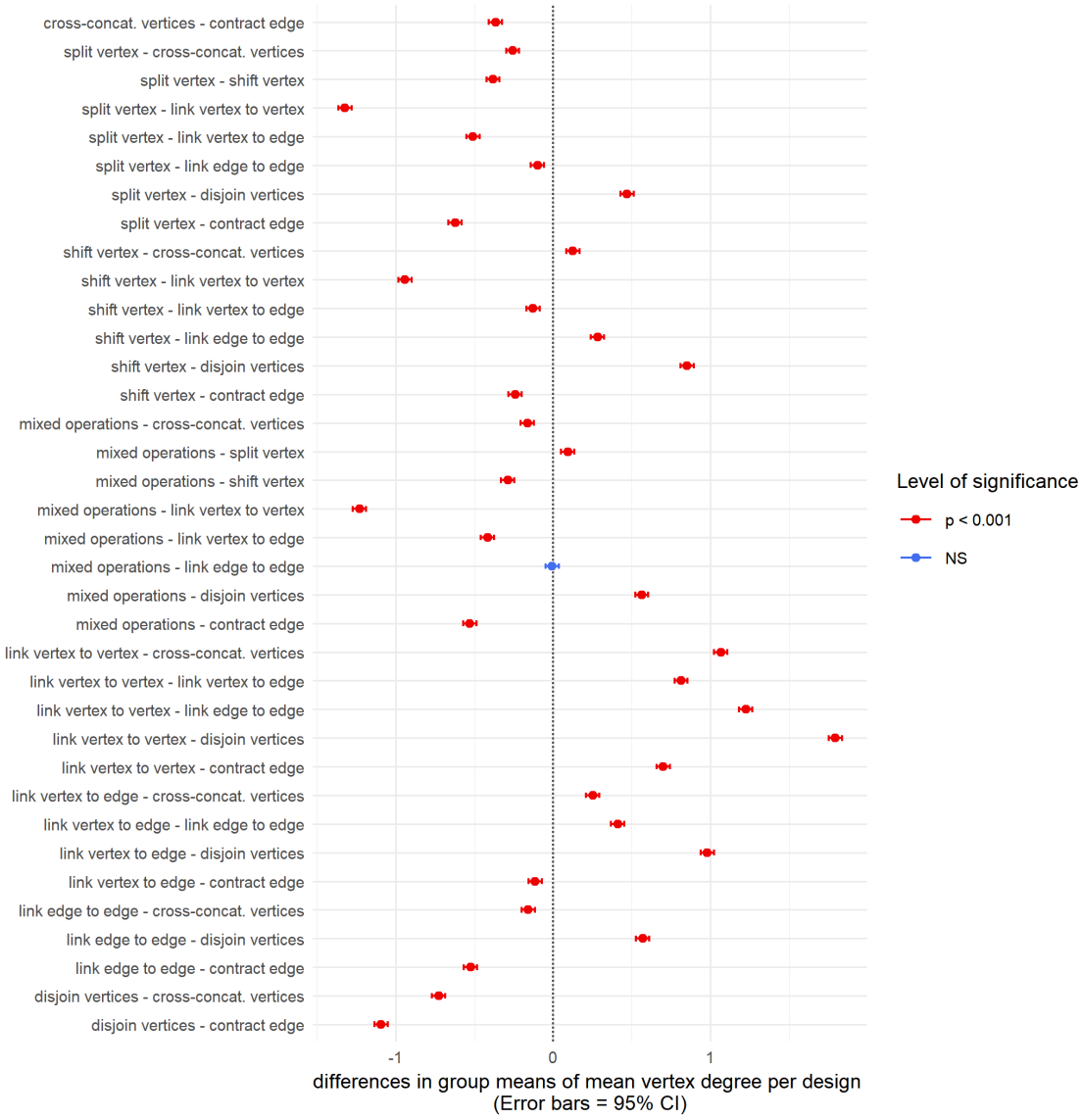

Figure 209. Differences between the group means of the mean vertex degree per design and the 95% confidence intervals (Tukey's HSD test).

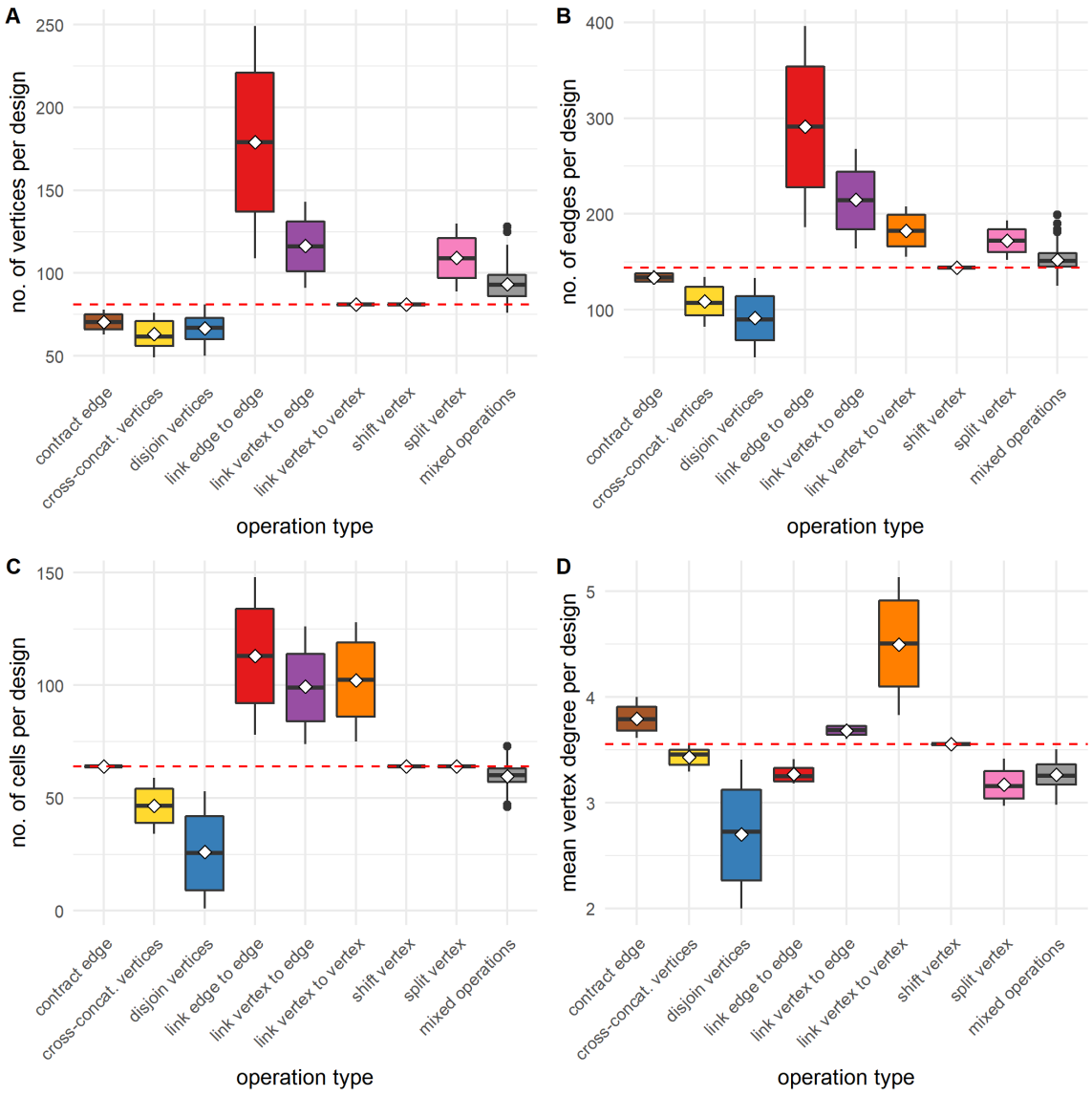

Figure 210. Boxplots showing the distribution of measurements related to the graph properties of a street graph. Measurements are grouped by the type of operation involved. In each boxplot, the red dashed line indicates the corresponding measurement for the initial 9 × 9 square-grid design.

Summary

The overall impacts on the graph properties can be summarized as follows.

1. Except for shifting vertex, all of the operations studied can change the number of vertices in a design and the mean vertex degree per design.
2. Certain types of operations maintain one or more of the graph properties: (a) by definition, shifting vertex does not change any of the graph properties for a design; (b) linking vertex to vertex maintains the number of vertices in a design; (c) contracting edge and splitting vertex maintain the number of cells in a design.
3. Linking vertex to edge is the only operation that causes an increase in all the four graph properties: it increases the numbers of vertices, edges, and cells in a design, and also increases the mean vertex degree per design.
4. The three linking operations all add new edges to a street graph, thus increasing its size. They also increase the number of cells in a design by splitting an existing cell in two.
5. Contracting edge, cross-concatenating vertices, and disjoining vertices reduce the number of edges in a design, thus reducing its graph size.
6. Among all the operations, only contracting edge, linking vertex to edge, and linking vertex to vertex definitely increase the mean vertex degree per design. All the rest tend to reduce the mean vertex degree per design.
7. The two operations, contracting edge and splitting vertex, make an interesting contrast: while contracting edges decrease the numbers of vertices and edges and increases the mean vertex degree per design, splitting vertex increases the numbers of vertices and edges and decreases the mean vertex degree per design.

8. Unlike the application of individual rules, the mixed operation has an inconsistent effect on the numbers of vertices, edges, and cells in a design.

10.1.2 Density: Streets, blocks, intersections, connectivity

(1) Total street length per design

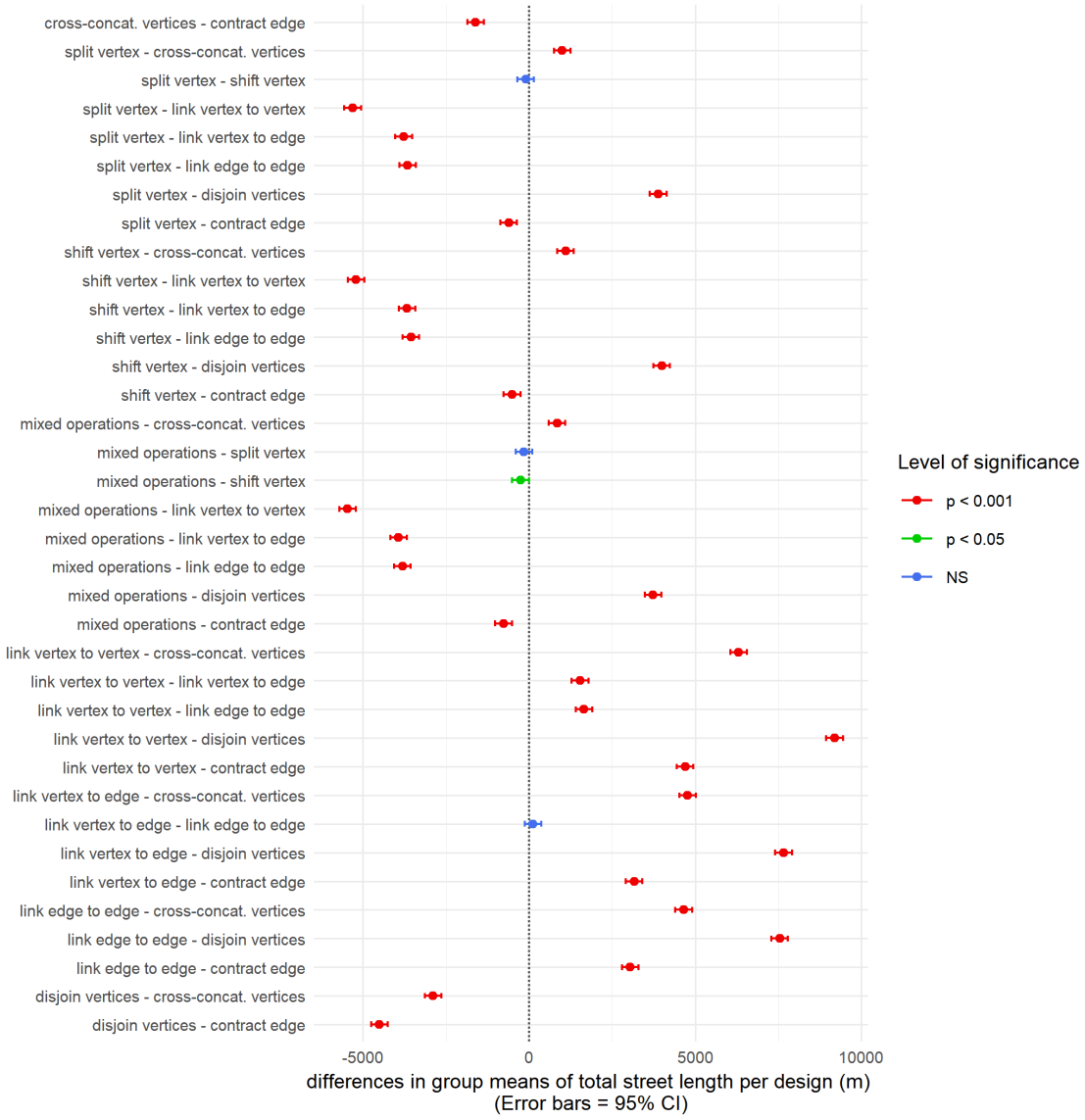

Figure 211. Differences between the group means of the total street length per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the total street length per design are significantly different from each other except between linking vertex to edge and linking edge to edge, between splitting vertex and shifting vertex, and between the mixed operations and splitting vertex (Figure 211). All the group means of the total street length per design are significantly different from the total street length for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 214A, the three linking operations can most dramatically increase the total street length in a design. On the other hand, the most effective operation to reduce the total street length per design is disjoining vertices.

(2) Total number of blocks per design

The total number of blocks per design is the same as the total number of cells per design. Therefore, Figure 214B is identical to Figure 210C.

(3) Total number of intersections per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the total number of intersections per design are significantly different from each other except between shifting vertex and linking vertex to vertex (Figure 212). There is no significant difference between shifting vertex and linking vertex to vertex on the measurement of the total number of intersections per design because shifting vertex never changes the number of intersections in a design and linking vertex to vertex only marginally increases the total number of vertices (when it links one of the corner vertices to convert it to an intersection) in the unlikely cases. Except shifting vertex, all the group means of the total number of intersections per design are significantly different from the total number of

intersections in the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 214C, linking edge to edge can dramatically increase the total number of intersections in a design, because each time we link an edge to another edge, we create two additional intersections.

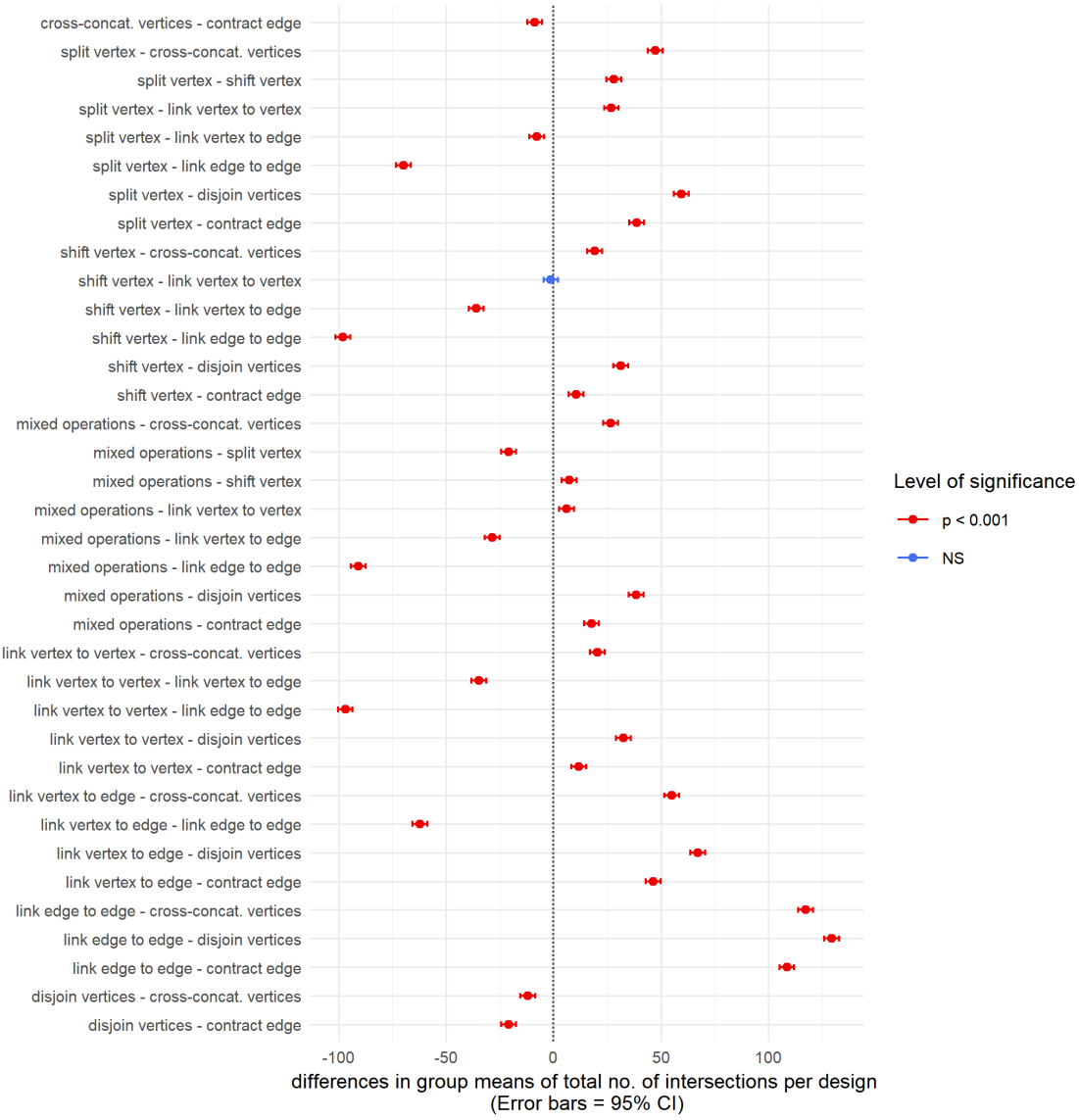

Figure 212. Differences between the group means of the total number of intersections per design and the 95% confidence intervals (Tukey's HSD test).

(4) Mean distance between intersections per design

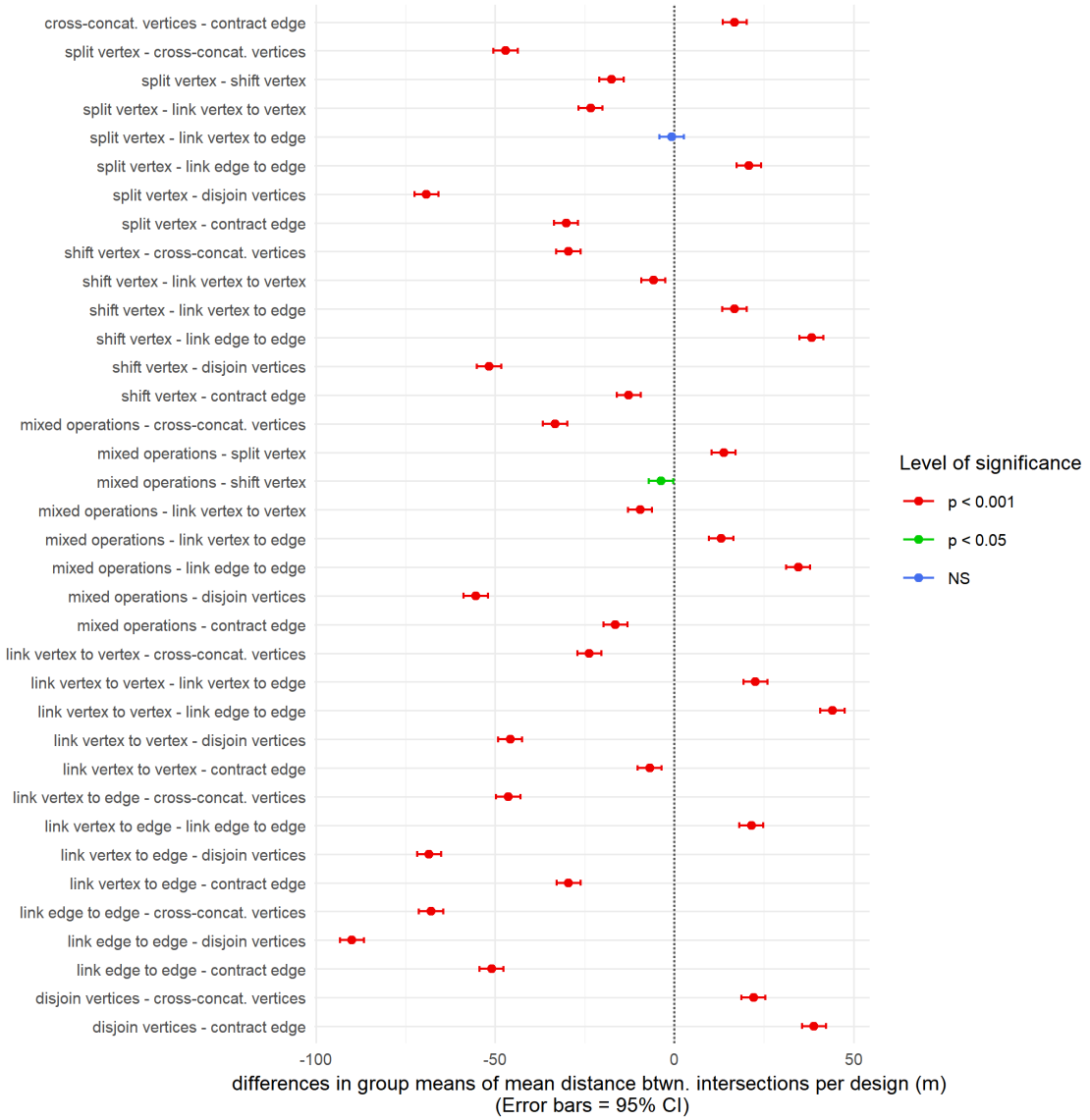

Figure 213. Differences between the group means of the mean distance between intersections per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the mean distance between intersections per design are significantly different from each other except between splitting vertex and linking vertex to edge (Figure 213). All the group

means are significantly different from the mean distance between intersections for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 214D, disjoining vertices can dramatically increase the mean distance between intersections per design. While linking edge to edge and linking vertex to edge decrease the mean distance between intersections per design, linking vertex to vertex increases it.

Summary

The overall impacts on the different density measures can be summarized as follows.

1. Certain types of operations have no impact on certain density measures: (a) contracting edge, shifting vertex, and splitting vertex maintain the number of blocks in a design; (b) shifting vertex maintains the number of intersections in a design.
2. Generally speaking, operations that tend to increase the total street length per design, the total number of blocks per design, and the total number of intersections per design also tend to decrease the mean distance between intersections per design, such as linking edge to edge and linking vertex to edge. Operations that tend to decrease the total street length per design, the number of blocks per design, and the number of intersections per design also tend to increase the mean distance between intersections per design, such as cross-concatenating vertices and disjoining vertices. However, there are exceptions. For example, while contracting edges tends to increase the total street length per design, it decreases the number of intersections per design and increases the mean distance

between intersections per design. On the other hand, while linking vertex to vertex increases the total street length and the number of blocks in a design, it tends to increase the mean distance between intersections slightly.

3. The three linking operations increase the total street length and the number of blocks per design much more dramatically than the other operations do. However, they have different impacts on the number of intersections per design: linking edge to edge increases the number of intersections at the fastest rate, followed by linking vertex to edge. By contrast, linking vertex to vertex almost has no impact on the number of intersections per design.
4. Splitting vertex can effectively increase the number of intersections per design and reduce the mean distance between intersections per design by consistently transforming a cross-intersection into two T-junctions. In doing so, it barely increases the total street length per design.
5. The mixed operations do not produce consistent effects on any of the density measures, and the resulting effect size is generally small compared to the other operations.

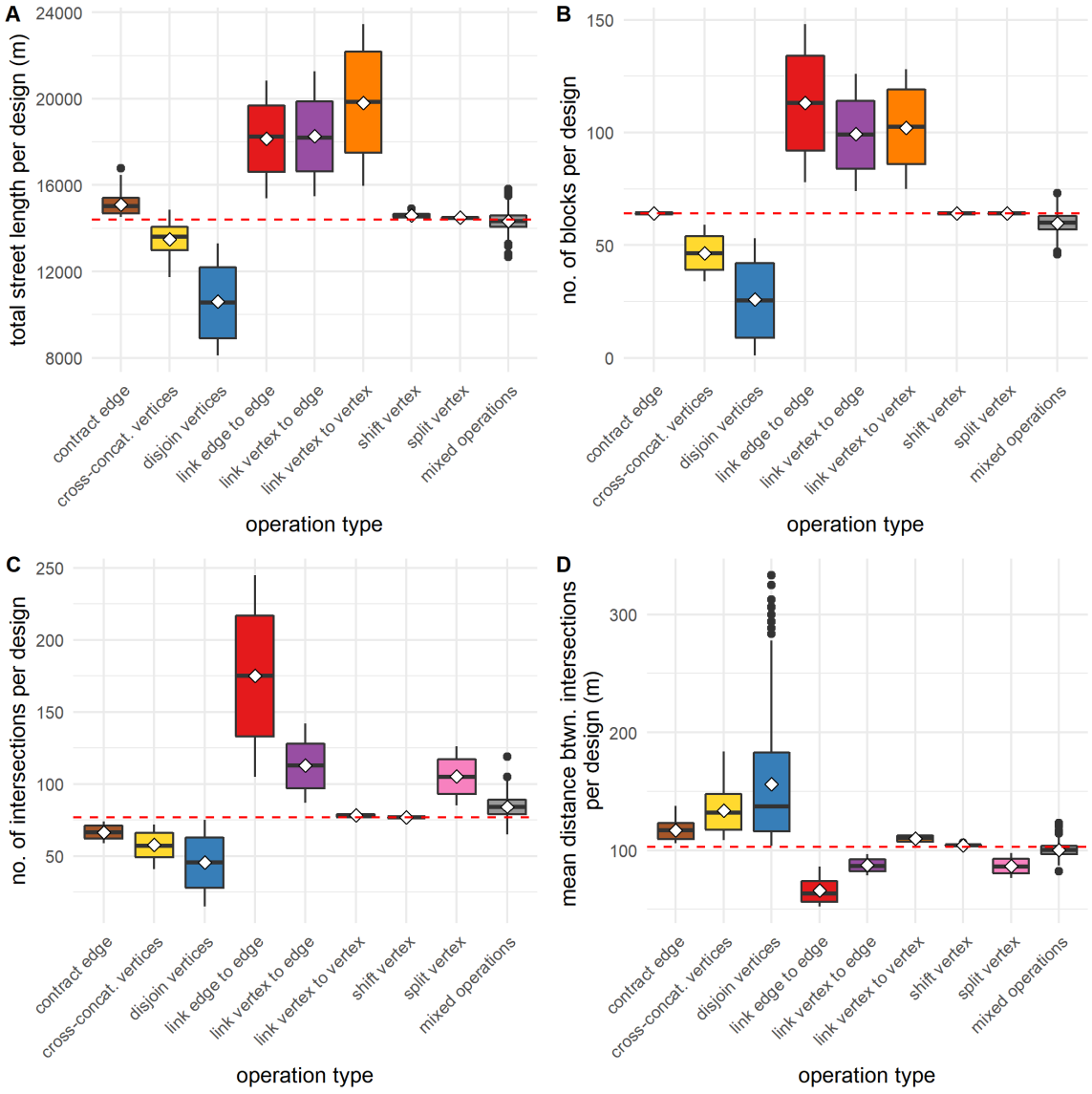

Figure 214. Boxplots showing the distribution of measurements related to the density of a street network. Measurements are grouped by the type of operation involved. In each boxplot, the red dashed line indicates the corresponding measurement for the initial 9 × 9 square-grid design.

10.1.3 Directional reach and directional distance

(1) Mean DDL20d per design

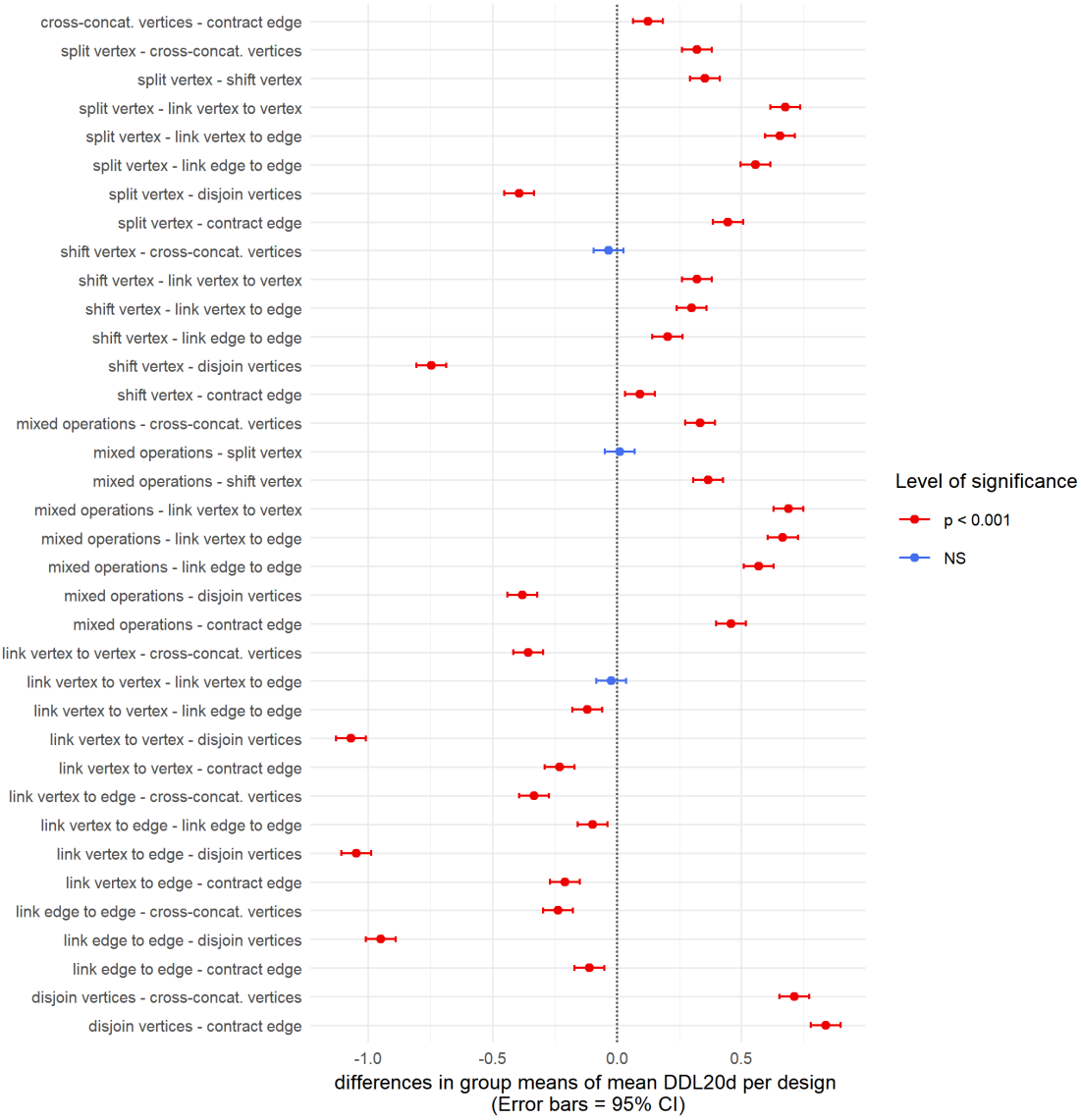

Figure 215. Differences between the group means of the mean DDL20d per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the mean DDL20d per design are significantly different from each other except between shifting

vertex and cross-concatenating vertices, between linking vertex to vertex and linking vertex to edge, and between the mixed operations and splitting vertex (Figure 215). All the group means are significantly higher than the mean DDL20d for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 218A, disjoining vertices most dramatically increased the mean DDL20d per design, followed by splitting vertex and the mixed operations. The three linking operations—linking edge to edge, linking vertex to edge, and linking vertex to vertex—had the least effects on the mean DDL20d per design.

(2) Mean linear reach per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the mean linear reach per design are significantly different from each other except between the operations of contracting edge, cross-concatenating vertices, and shifting vertex, between splitting vertex and disjoining vertices, between linking vertex to edge and linking edge to edge, and between linking vertex to vertex and linking vertex to edge (Figure 216). All group means are significantly lower than the mean linear reach for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 218B, applying the mixed operations can most dramatically decrease the mean linear reach per design, followed by splitting vertex and disjoining vertices. Among all the groups of designs, only a few designs generated by the operation of cross-concatenating vertices have a mean linear reach that is greater than the mean linear reach for the initial design.

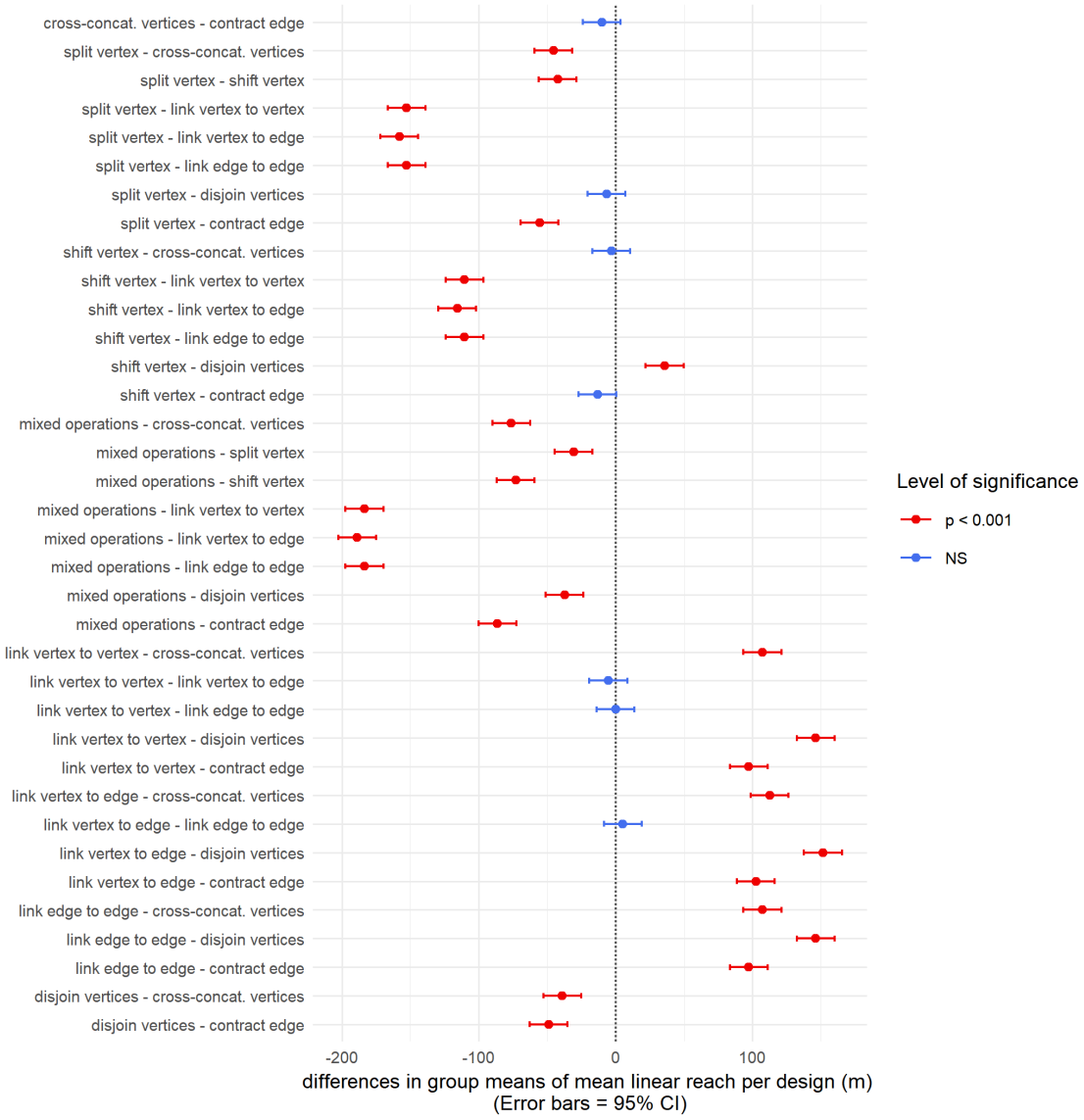

Figure 216. Differences between the group means of the mean linear reach per design and the 95% confidence intervals (Tukey's HSD test).

(3) Mean 2-dc reach per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the mean 2-dc reach per design are significantly different from each other only except between the mixed operations and splitting vertex (Figure 217).

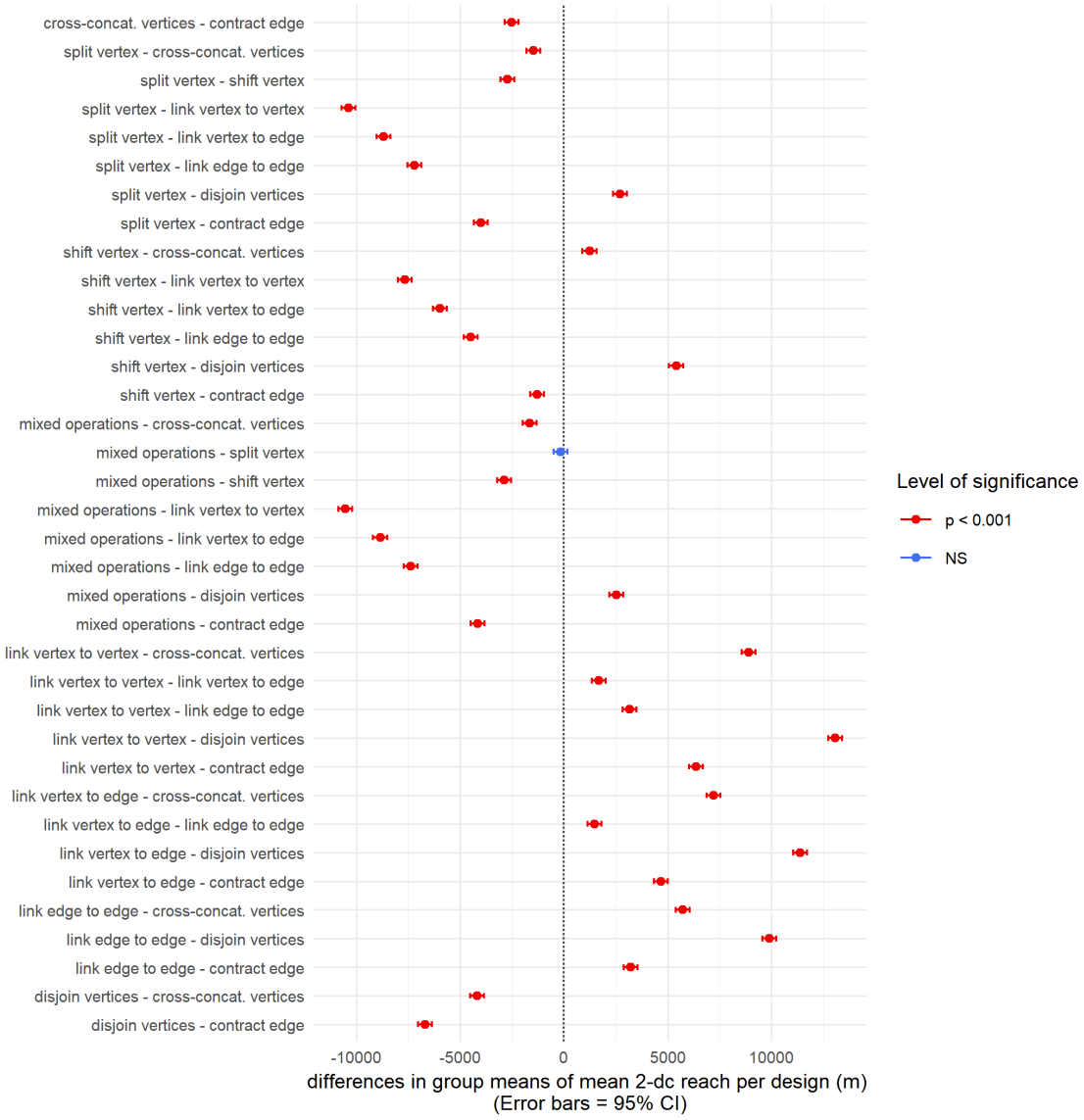

Figure 217. Differences between the group means of the mean 2-dc reach per design and the 95% confidence intervals (Tukey's HSD test).

The group means for the operations of contracting edge, cross-concatenating vertices, disjoining vertices, shifting vertex, splitting vertex, and the mixed operations are significantly lower than the mean 2-dc reach for the initial design (One-sample Student's *t*-test, $p < .001$). The group means for the operations of linking edge to edge, linking vertex to edge, and linking vertex to vertex are significantly greater than the mean 2-dc

reach for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 218C, disjoining vertices can dramatically decrease the mean 2-dc reach per design, as can splitting vertex and the mixed operations. On the other hand, linking vertex to vertex can dramatically increase the mean 2-dc reach per design.

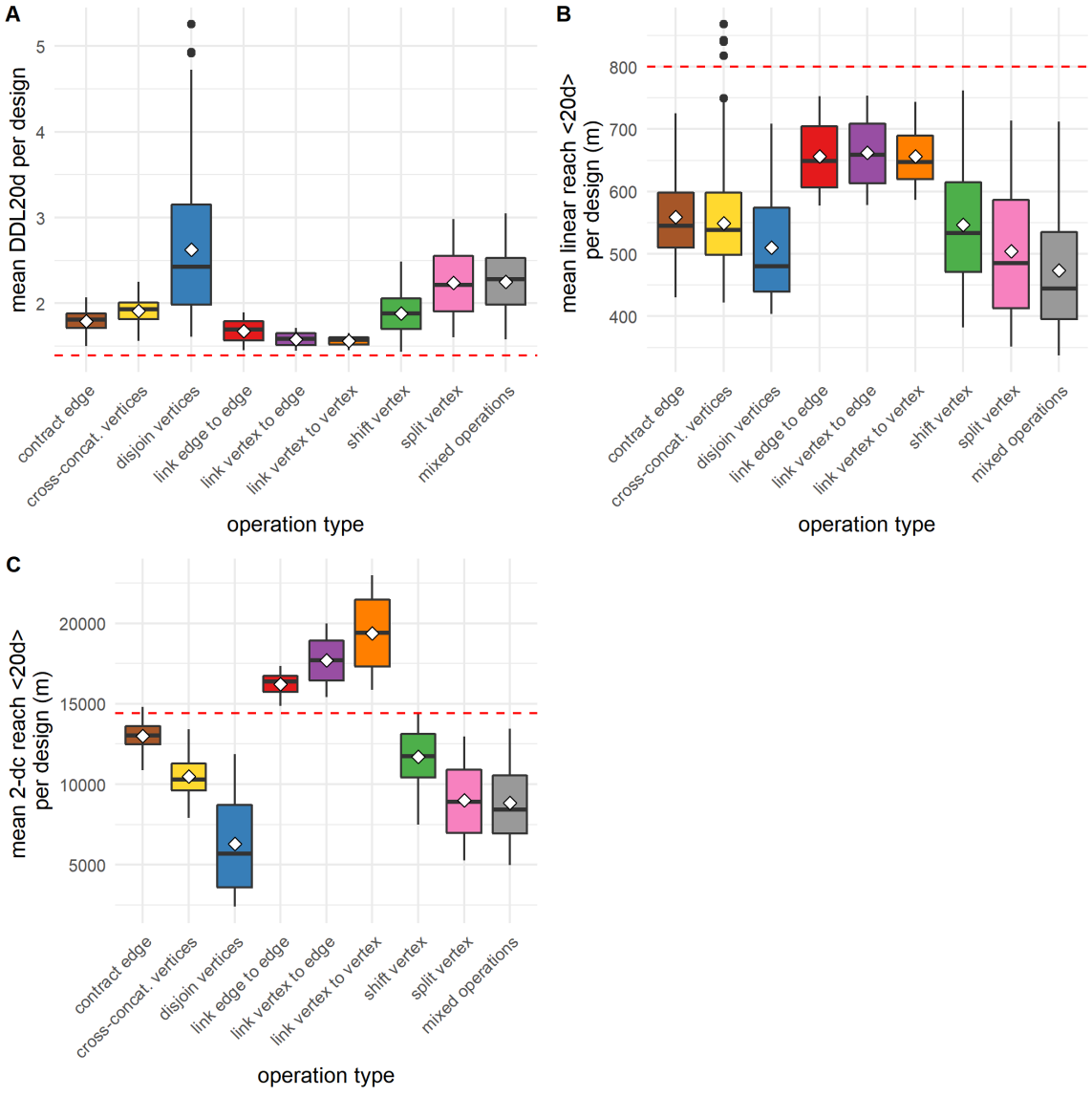

Figure 218. Boxplots showing the distribution of measurements related to the directional distance. Measurements are grouped by the type of operation involved. In each boxplot, the red dashed line indicates the corresponding measurement for the initial 9 × 9 square-grid design.

Summary

The overall impacts on the mean directional reach and directional distance can be summarized as follows.

1. All the operations had a significant impact on all the measures related to directional distance.
2. All the operations tend to increase the mean DDL20d per design.
3. All the operations tend to decrease the mean linear reach per design. Cross-concatenating vertices is the only operation that can potentially generate designs that score better at the mean linear reach than the initial square grid design.
4. Except for the three linking operations, all the operations tend to increase the mean DDL20d per design and decrease both the mean linear reach per design and the mean 2-dc reach per design. Like the other operations, the three linking operations increase the mean DDL20d per design and decrease the mean linear reach per design. However, unlike the other operations, the three linking operations tend to increase the mean 2-dc reach per design, partly because they all substantially increase the street density in a design.
5. Disjoining vertices had the greatest effect on the mean DDL20d per design. It can produce extremely segregated designs. Splitting vertex and the mixed operations also tend to increase the mean DDL20d per design quickly.
6. Among all the operations, the three linking operations had the least effect on the mean DDL20d per design.

10.1.4 Geometric Regularity

(1) Fragmentality per design

Figure 219. Differences between the group means of fragmentality per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the fragmentality per design are significantly different from each other except between

shifting vertex and contracting edge, between disjoining vertices and cross-concatenating vertices, between linking vertex to edge and linking edge to edge, and between linking vertex to vertex and linking vertex to edge (Figure 219). All the group means are significantly higher than the fragmentality for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 223A, the designs with the highest fragmentality were generated by the operation of disjoining vertices. Cross-concatenating vertices and applying the mixed operations can also generate designs with a high fragmentality. By contrast, the three linking operations had relatively low effects in increasing a design's fragmentality.

(2) Standard deviation of block area per design

Only about half of all pairwise comparisons (Tukey's HSD test) suggest a significant difference exists between the group means of the standard deviation of the block area per design (Figure 220). Nevertheless, all group means are significantly greater than zero (One-sample Student's *t*-test, $p < .001$). As shown in Figure 220 and Figure 223B, disjoining vertices can substantially increase the standard deviation of the block area per design, and it produced the greatest effect.

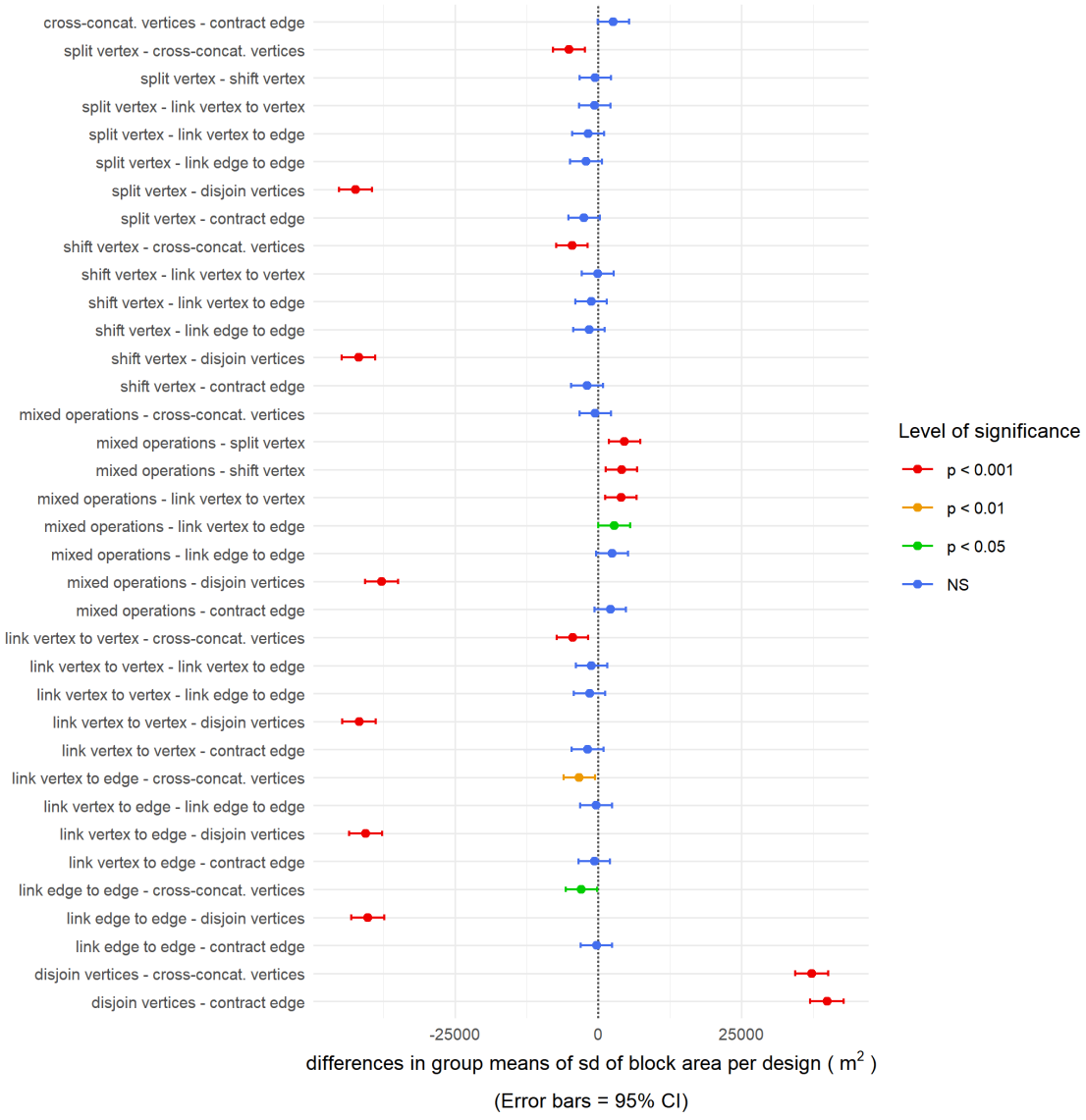

Figure 220. Differences between the group means of the standard deviation of the block area per design and the 95% confidence intervals (Tukey's HSD test).

(3) Standard deviation of block perimeter per design

Over half of all pairwise comparisons (Tukey's HSD test) suggest a significant difference exists between the group means of the standard deviation of the block perimeter per design (Figure 221). All group means are significantly greater than zero

(One-sample Student's *t*-test, $p < .001$). As shown in Figure 221 and Figure 223C, disjoining vertices can substantially increase the standard deviation of the block perimeter per design, and it produced the greatest effect.

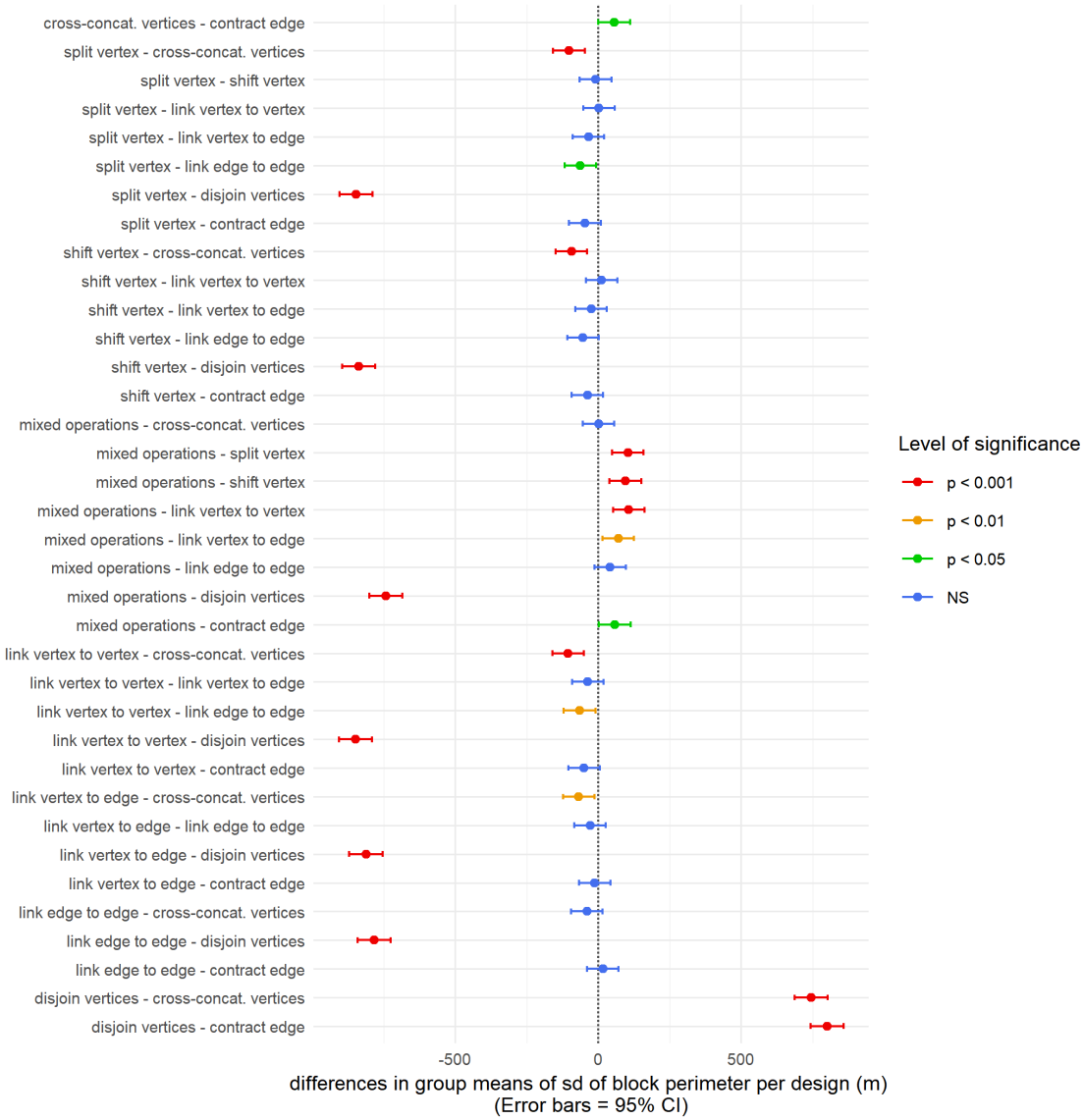

Figure 221. Differences between the group means of the standard deviation of the block perimeter per design and the 95% confidence intervals (Tukey's HSD test).

(4) Mean SAPR per design

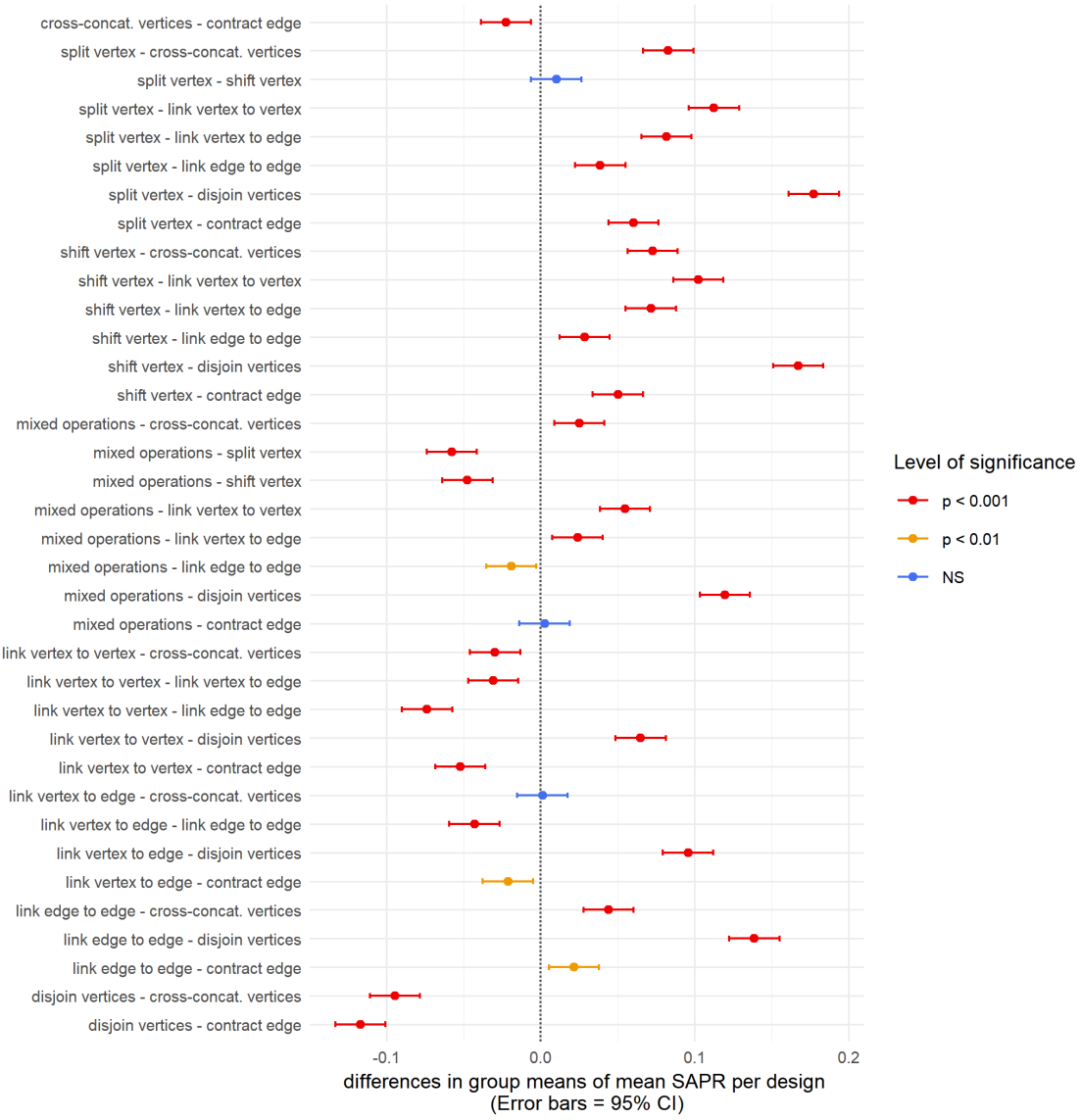

Figure 222. Differences between the group means of the mean SAPR per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the mean SAPR per design are significantly different from each other except between splitting vertex and shifting vertex, between the mixed operations and contracting edge, and

between linking vertex to edge and cross-concatenating vertices. All group means are less than the mean SAPR for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 223D, the designs that have the lowest mean SAPRs were generated by disjoining vertices or cross-concatenating vertices.

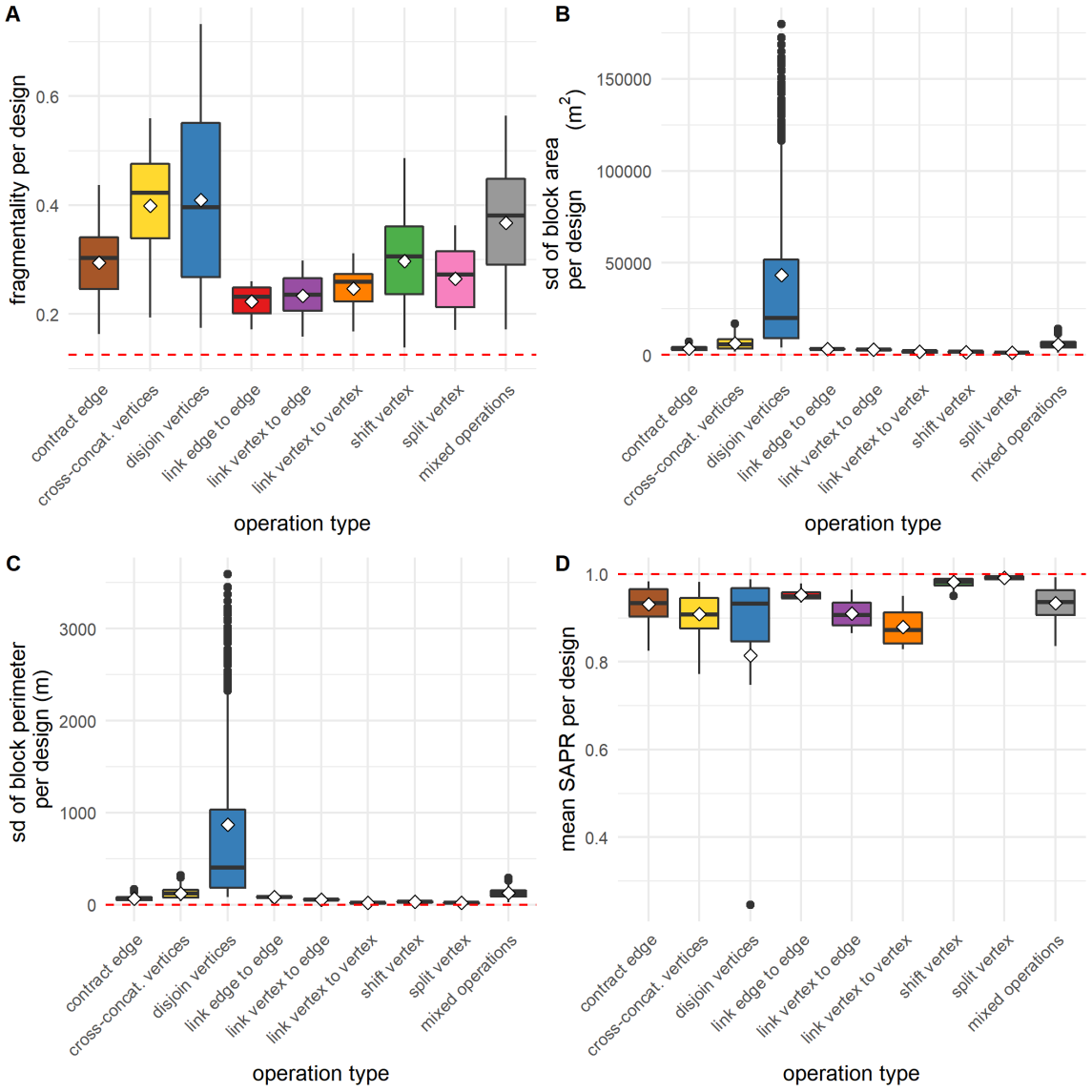

Figure 223. Boxplots showing the distribution of measurements related to the regularity of a street network. Measurements are grouped by the type of operation involved. In each boxplot, the red dashed line indicates the corresponding measurement for the initial 9 × 9 square-grid design.

Summary

The overall impacts on the regularity of a street network can be summarized as follows.

1. Clearly, all the operations tend to make designs more irregular, based on the measures studied here.
2. Disjoining vertices can produce extremely irregular designs. It also produced the widest range of designs in terms of regularity.
3. The operations that produced the greatest effect on the fragmentality of designs are cross-concatenating vertices, disjoining vertices, and the mixed operations, followed by contracting edge and shifting vertex. They all tend to increase the fragmentality per design.
4. Excluding the operation of disjoining vertices, the greatest effect on the differentiation of block size (indicated by the area and the perimeter of a block) was caused by cross-concatenating vertices and the mixed operations.
5. Splitting vertex and shifting vertex had the least effects on the mean SAPR per design.
6. Overall, disjoining vertices had the greatest effect on the mean SAPR per design, followed by the linking vertex to vertex, linking vertex to edge, and cross-concatenating vertices.

10.1.5 Diversity in syntactic conditions

(1) Total number of distinct DDL20d values per design

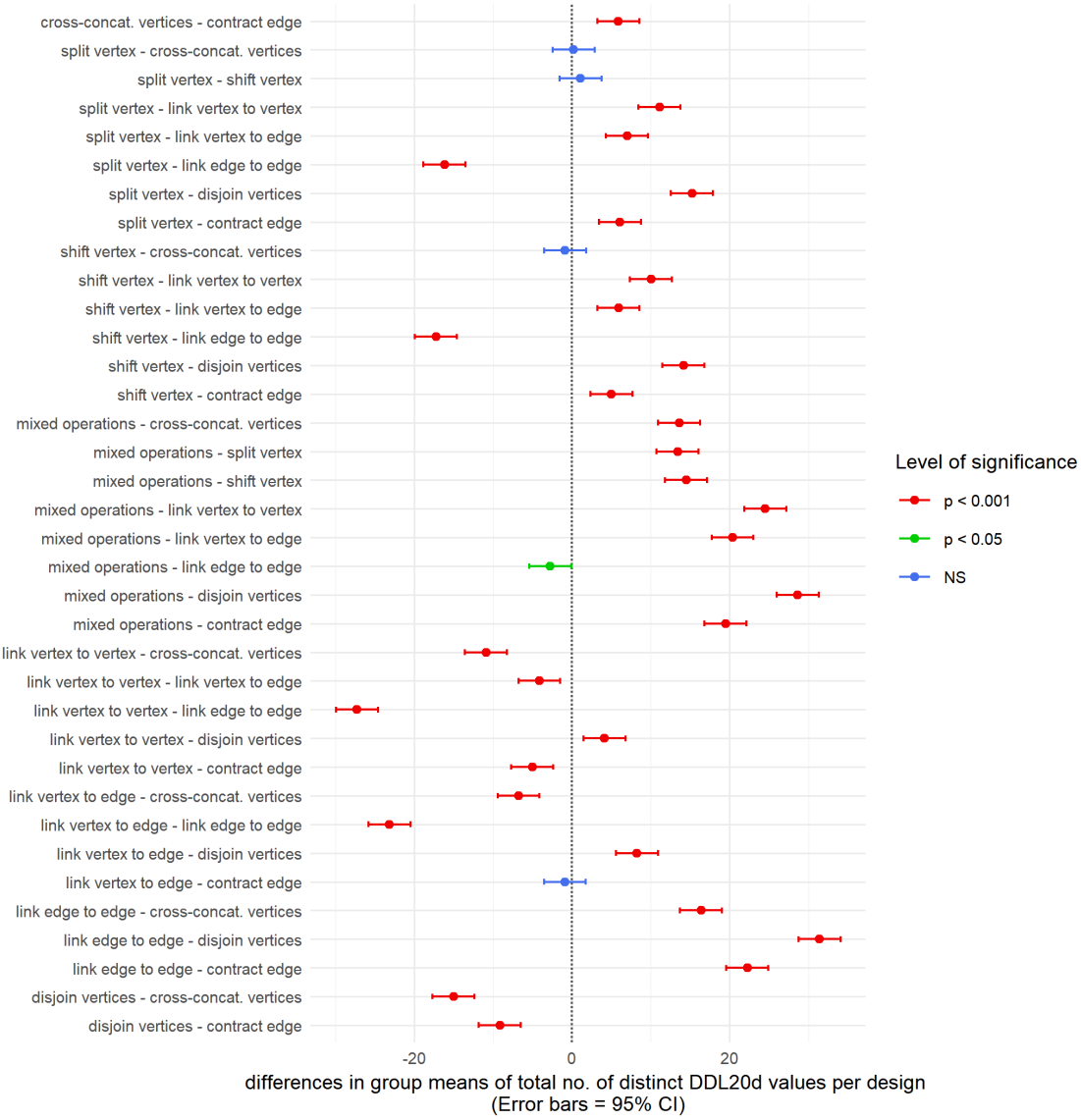

Figure 224. Differences between the group means of the total number of distinct DDL20d values per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the total number of distinct DDL20d values per design are significantly different from each other

except between splitting vertex and cross-concatenating vertices, between splitting vertex and shifting vertex, between shifting vertex and cross-concatenating vertices, and between linking vertex to edge and contracting edge (Figure 224). All group means are significantly higher than the total number of distinct DDL20d values in the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 227A, designs with the greatest number of distinct DDL20d values were generated by linking edge to edge and the mixed operations.

(2) Proportion of distinct DDL20d values per design

Pairwise comparisons (Tukey's HSD test) show that the group means of the proportion of distinct DDL20d values per design are significantly different from each other except between shifting vertex and contracting edge and between linking vertex to vertex and linking vertex to edge (Figure 225). All group means are significantly higher than the proportion of distinct DDL20d values for the initial design (One-sample Student's *t*-test, $p < .001$). As shown in Figure 227B, unlike the measurement of the total number of distinct DDL20d values, the measurement of the proportion of distinct DDL20d values per design counteracts the effect of the size of the design. For example, although linking edge to edge can generate designs that have a great number of distinct DDL20d values, it performs relatively poorly if the diversity is evaluated by the proportion instead of the absolute number of distinct DDL20d values. The designs that have the greatest proportion of distinct DDL20d values were generated by disjoining vertices, cross-concatenating vertices, and the mixed operations.

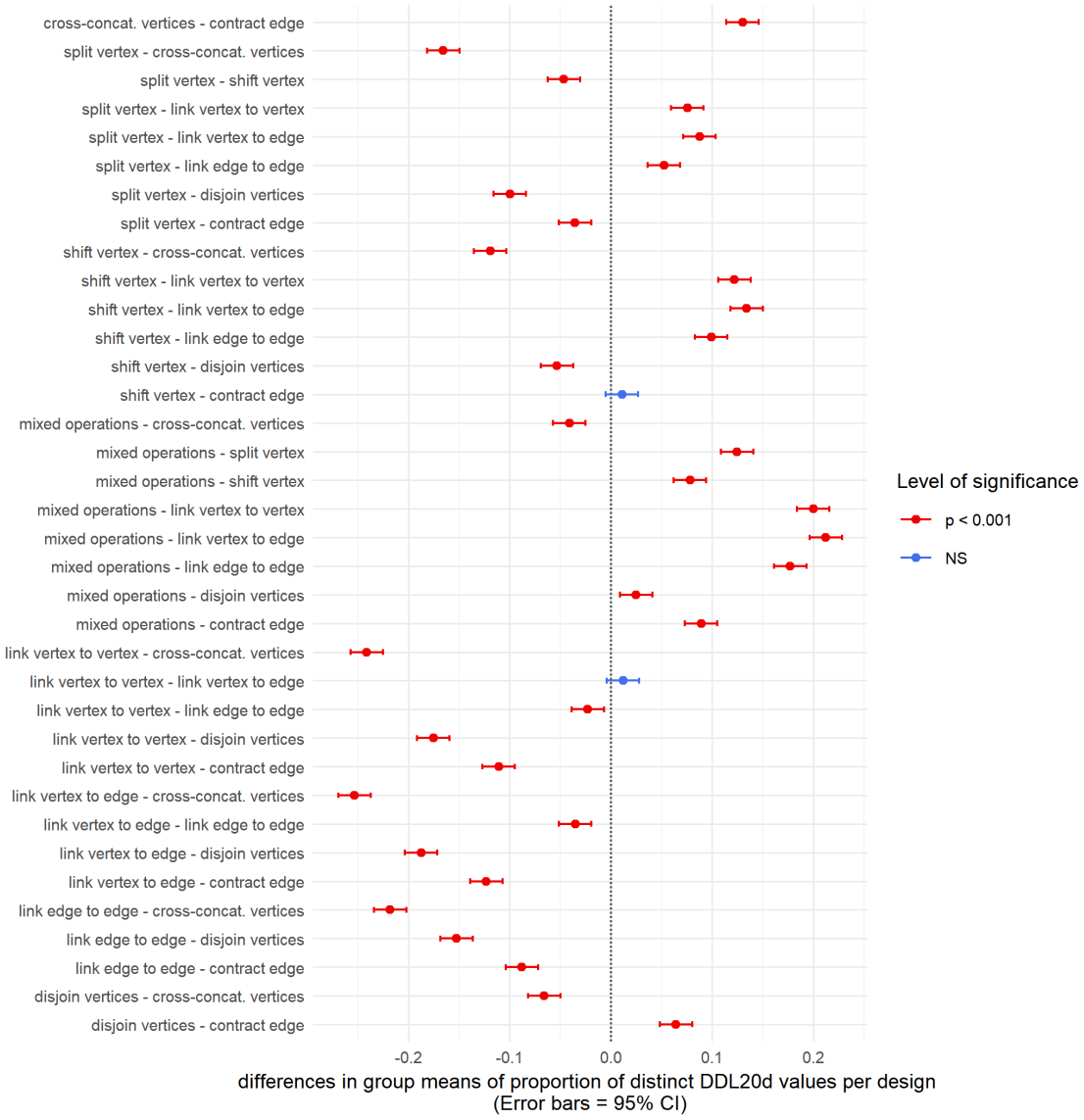

Figure 225. Differences between the group means of the proportion of distinct DDL20d values per design and the 95% confidence intervals (Tukey's HSD test).

(3) Standard deviation of DDL20d values per design

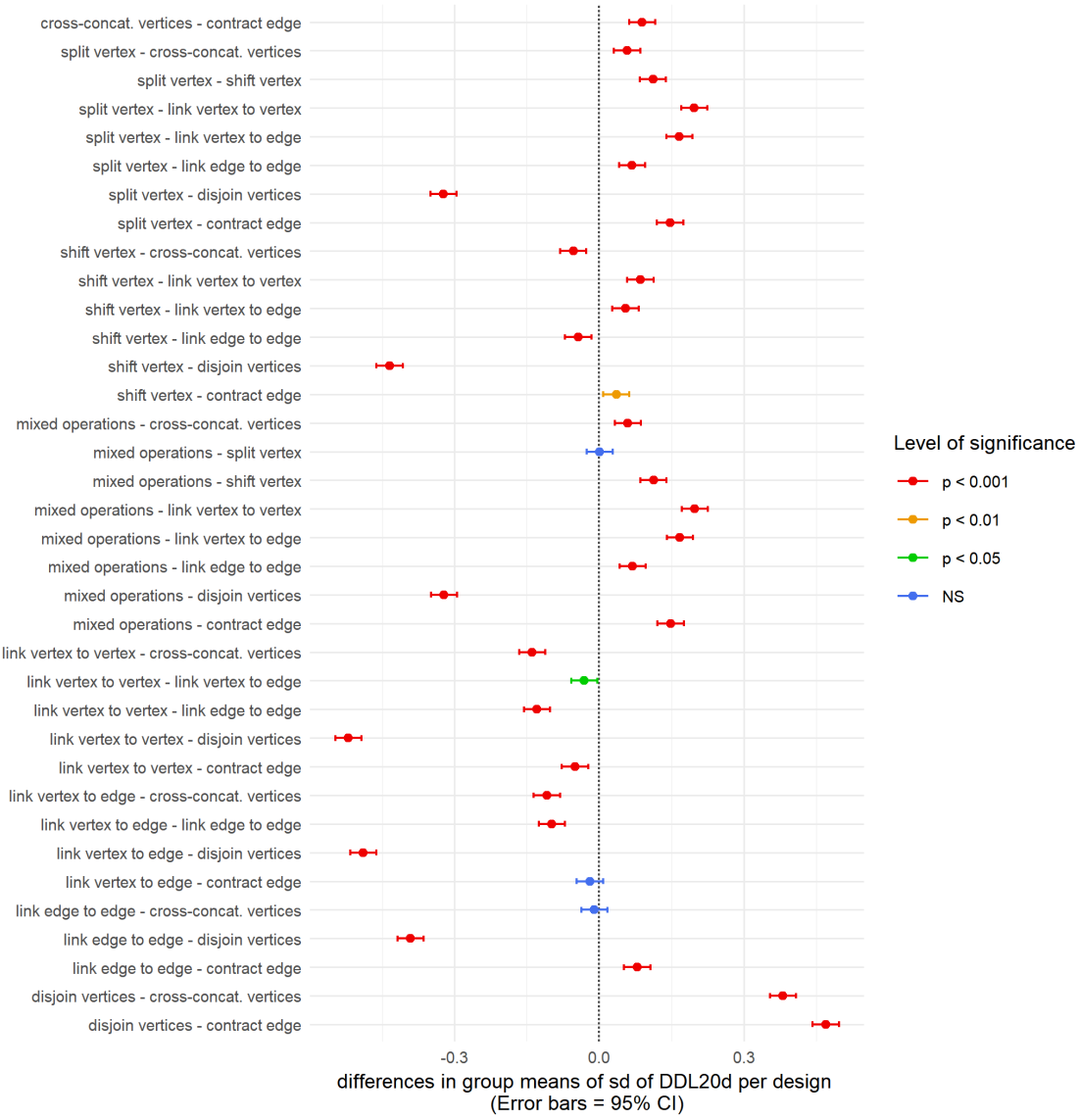

Figure 226. Differences between the group means of the standard deviation of DDL20d per design and the 95% confidence intervals (Tukey's HSD test).

Pairwise comparisons (Tukey's HSD test) show that the group means of the standard deviation of DDL20d per design are significantly different from each other except between the mixed operations and splitting vertex, between linking vertex to edge

and contracting edge, and between linking edge to edge and cross-concatenating vertices (Figure 226). All group means are significantly greater than zero (One-sample *t*-test, $p < .001$). As shown in Figure 227C, disjoining vertices can most dramatically increase the standard deviation of DDL20d per design. The other operations are more or less comparable, among which the mixed operations and splitting vertex had greater effects on the standard deviation of DDL20d per design.

Summary

The overall impacts on the syntactic diversity of a street network can be summarized as follows.

1. All the operations can produce different syntactic conditions in a design.
2. Cross-concatenating vertices had the greatest effect on the proportion of unique DDL20d values in a design, followed by the mixed operations and disjoining vertices. They all increased the proportion of distinct DDL20d values in a design.
3. The three linking operations had the least effects on the proportion of unique DDL20d values in a design.
4. After relativizing the count of unique DDL20d values against the total number of segments in a design, the relative position in the boxplot has changed dramatically for the operations of linking edge to edge, cross-concatenating vertices, and disjoining vertices. While linking edge to edge can generate many different syntactic conditions in absolute quantity, it also rapidly increases the total number of segments in the design. Therefore, the proportion of distinct DDL20d values is low. By contrast, while disjoining vertices tends to generate the fewest number of

distinct syntactic conditions, the proportion is still quite high because the total number of segments in a design is reduced.

5. Disjoining vertices had the greatest effect on the standard deviation of DDL20d per design.

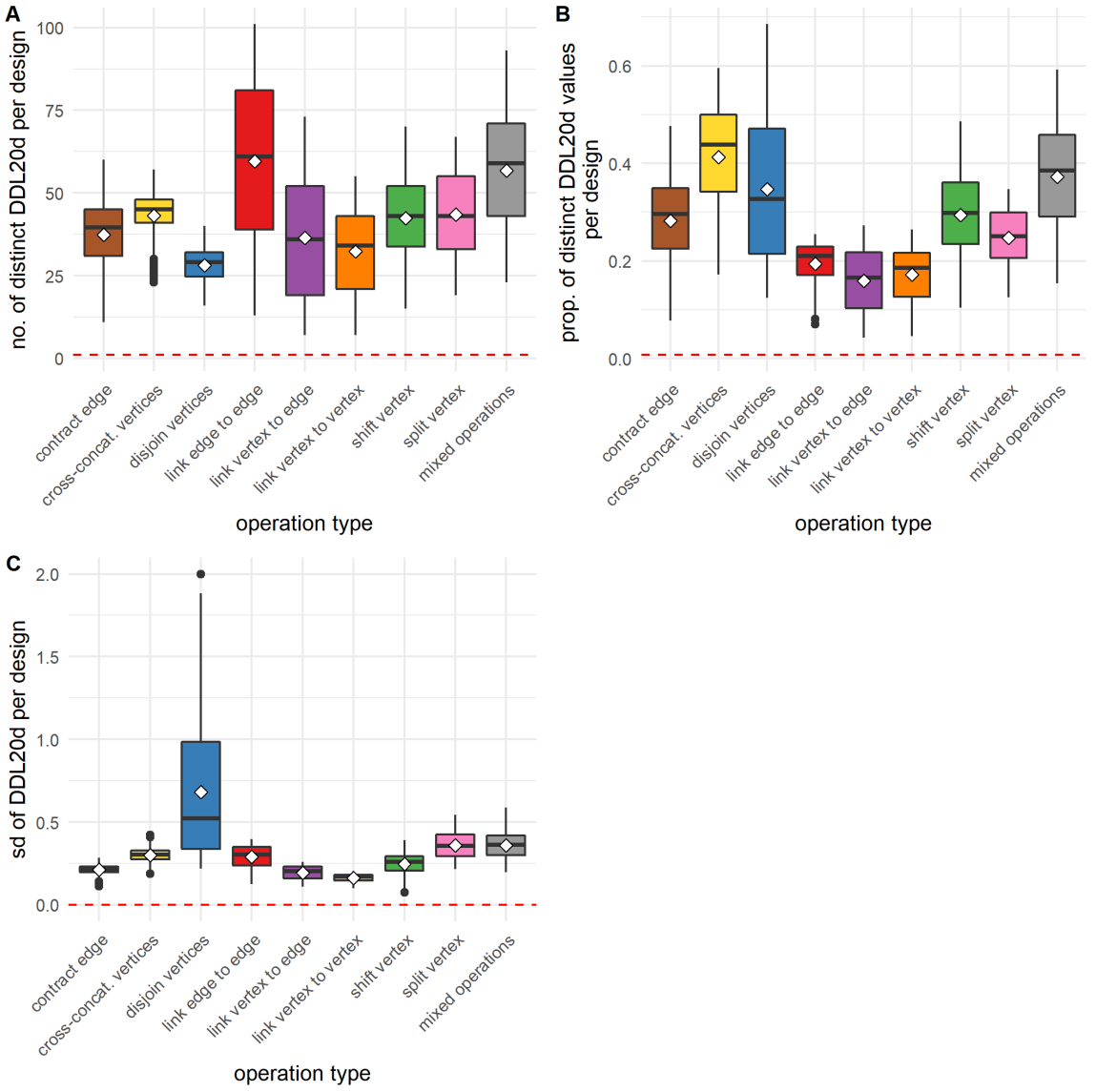

Figure 227. Boxplots showing the distribution of measurements related to the diversity of syntactic conditions in a street network. Measurements are grouped by the type of operation involved. In each boxplot, the red dashed line indicates the corresponding measurement for the initial 9 × 9 square-grid design.

10.2 Comparison Based on the Relationship Between Variables

In the previous section, we sorted designs into different groups based on the specific type of operation used to generate the design, and then compared the group means across different measures. This comparison is useful in that it gives us a sense of the operations' different capacities to influence individual measures. However, we cannot attribute the difference between the group means entirely to the specific type of operation that has been used, because the difference in the frequency of applications of operations could also have played a role. For example, when generating designs by contracting edges, we never applied the operation of contracting edge more than 18 times. By contrast, when generating designs by linking edges to edges, we applied the operation of linking edge to edge up to 84 times. Therefore, to know how "efficiently" we can change the individual measures by applying each type of operation, we have to factor in the frequency of application of the operation. We can, of course, try to quantify the relationship between each measure that we have studied and the frequency of application of operation, and then compare how the relationships might be different for different types of operations. However, the conclusion would do little to inform design practice because few would conceive the design process as a sequential application of the different operations studied here. A more interesting question, then, is to seek the relationship between measures that are not tied to the specific operations studied here but are of more general interest.

In this regard, the measure of fragmentality per design can serve as a common ground to build our understanding of how different types of operations influence the various aspects of a design differently. Below we first study how fragmentality per design

varies with the frequency of application of an operation. Then we quantify the relationship between fragmentality per design and the proportion of distinct DDL20d values per design, the standard deviation of DDL20d per design, the mean DDL20d per design, and the mean SAPR per design, with simple linear regression models.

10.2.1 Fragmentality per design vs. frequency of application of operation (*T*)

For each group of designs, a Pearson correlation coefficient is computed to assess the relationship between the fragmentality per design and the frequency of application of operation. In all cases, there is a strong positive correlation between the two variables (Figure 228). However, the slopes of the regression lines are different from each other. For example, contracting edge can increase the fragmentality very quickly, as can cross-concatenating vertices. Linking edge to edge, on the other hand, increases the fragmentality at a much slower pace (Figure 229).

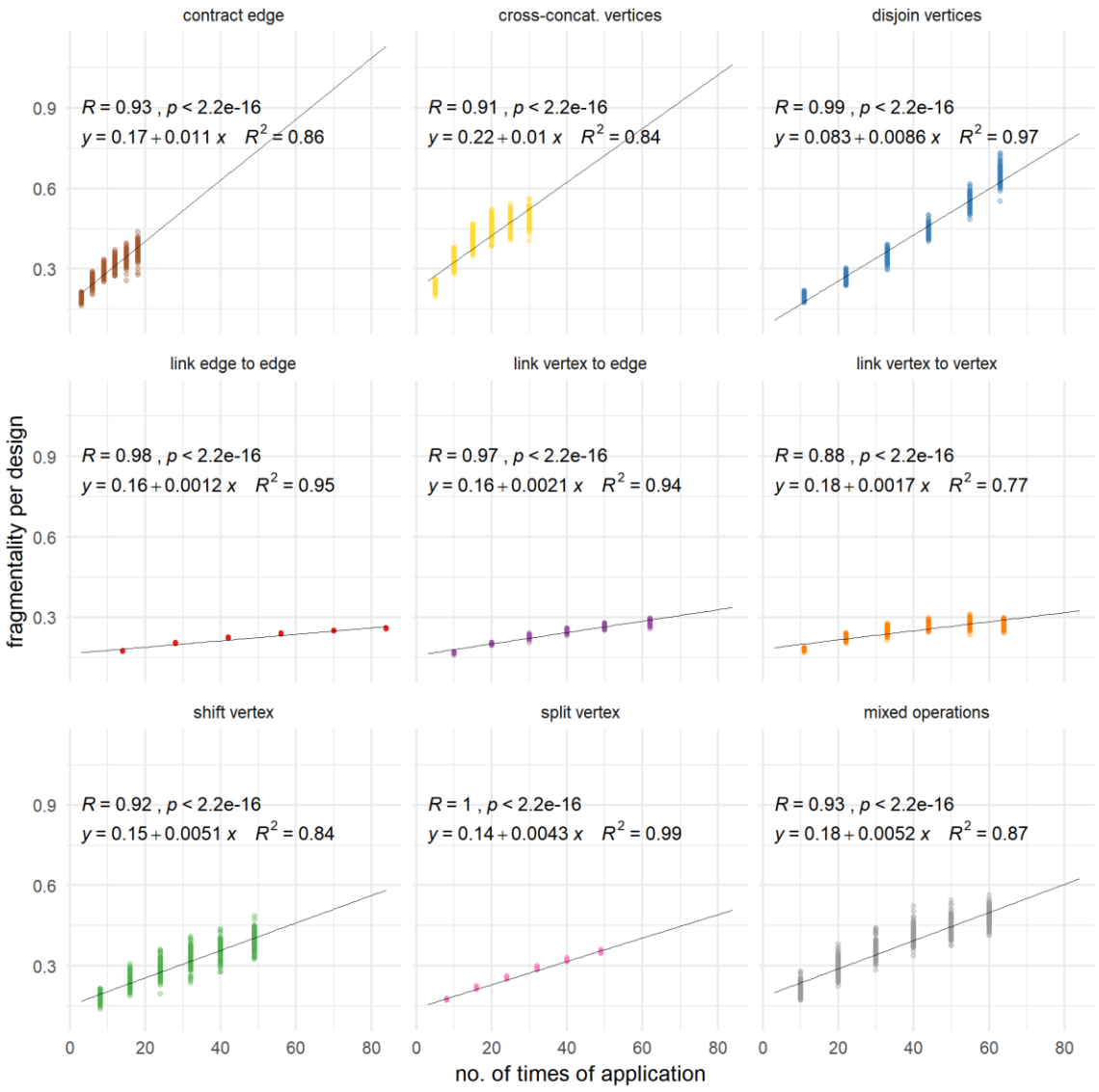

Figure 228. Scatterplots showing the relationship between fragmentality per design and frequency of application of operation. Black lines are linear least squares regression fits to data points. In each plot, the regression line is extended to span the full range of the plot for better visibility.

Although we can increase the fragmentality by applying any type of operation studied here, the implications of the increased fragmentality, from a designer's point of view, are quite different. For example, both shifting vertex and splitting vertex tend to increase the fragmentality per design at a similar pace (the slopes and intercepts of the

two least-squares lines are very similar), but they imply very different (if not opposite) design intentions.

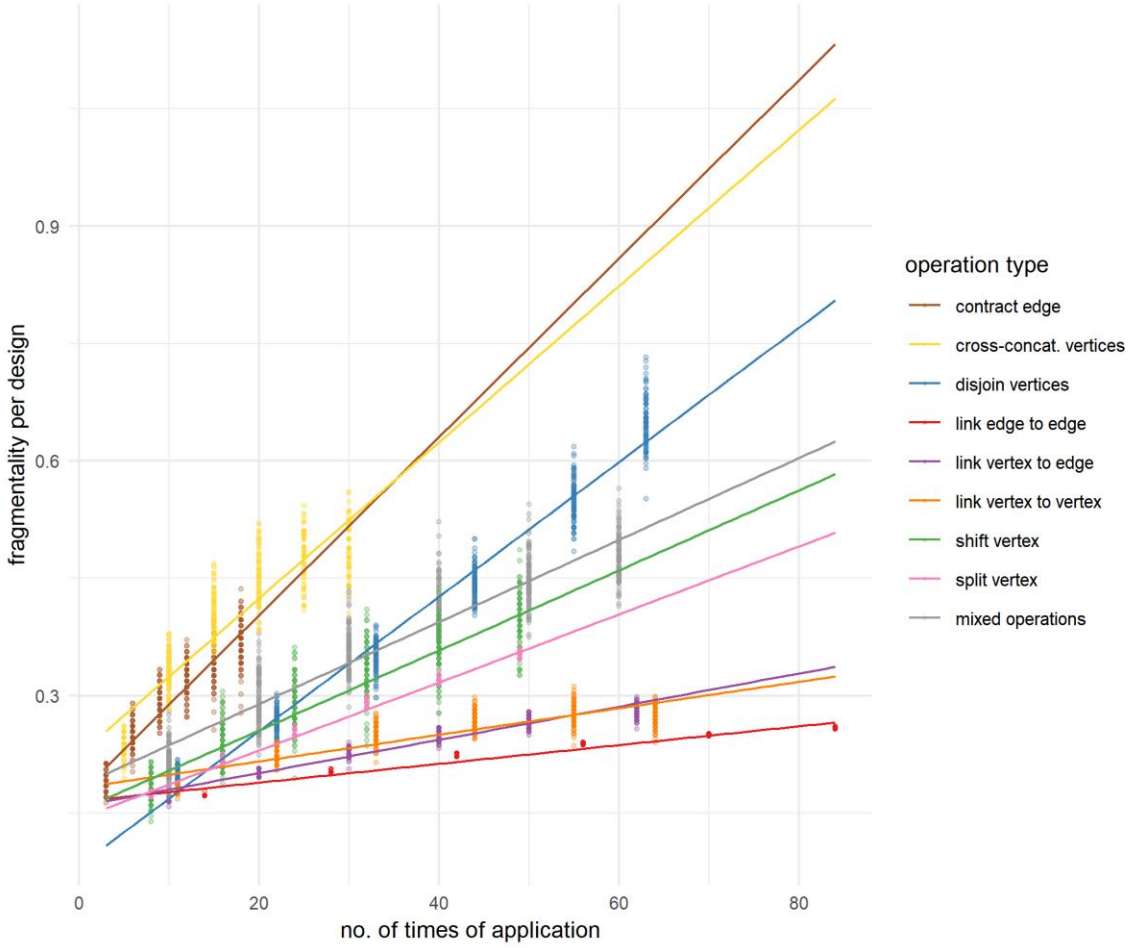

Figure 229. A scatterplot showing the relationship between fragmentality per design and frequency of application of operation for each group (operation type). Linear regression lines are extended to span the full range of the plot for better visibility.

Splitting vertex increases the fragmentality of a design by offsetting intersections and adding hard turns along existing streets. In doing so, it multiplies T-junctions in the street network, thus constantly prompting people to make binary choices (i.e., a left-turn or a right-turn). By contrast, shifting vertex preserves the topology of the existing street

structure and never reduces the choices of path at an intersection. The fragmentality is increased by gently varying the angular rotation of streets instead of drastically disrupting the linear flow of paths.

10.2.2 Proportion of distinct DDL20d values per design vs. fragmentality per design

For each group of designs, a Pearson correlation coefficient is computed to assess the relationship between the proportion of distinct DDL20d values per design and the fragmentality per design. In all cases, there is a strong positive correlation between the two variables (Figure 230). Further linear regression analysis shows that the slopes of the least-squares lines are fairly similar to each other, ranging from 0.92 to 1.6 (Figure 230 and Figure 231).

While the two variables covary with each other in a surprisingly consistent manner across the different groups of designs generated by different operations, it is not surprising that the proportion of distinct DDL20d values per design correlated so well with the fragmentality per design. The fragmentality per design, by definition, is a function of the number of continuity lines (that can be potentially formed) and the total number of segments in the street network, while the proportion of distinct DDL values, by definition, is a function of the number of unique DDL values and the total number of segments in the street network. Typically, the segments that lie on the same continuity line also share the same DDL value. Therefore, as long as the continuity lines bear different DDL values (which is frequently the case), the total number of continuity lines that can be formed in the street network is a very good predictor of the total number of distinct DDL values in the design.

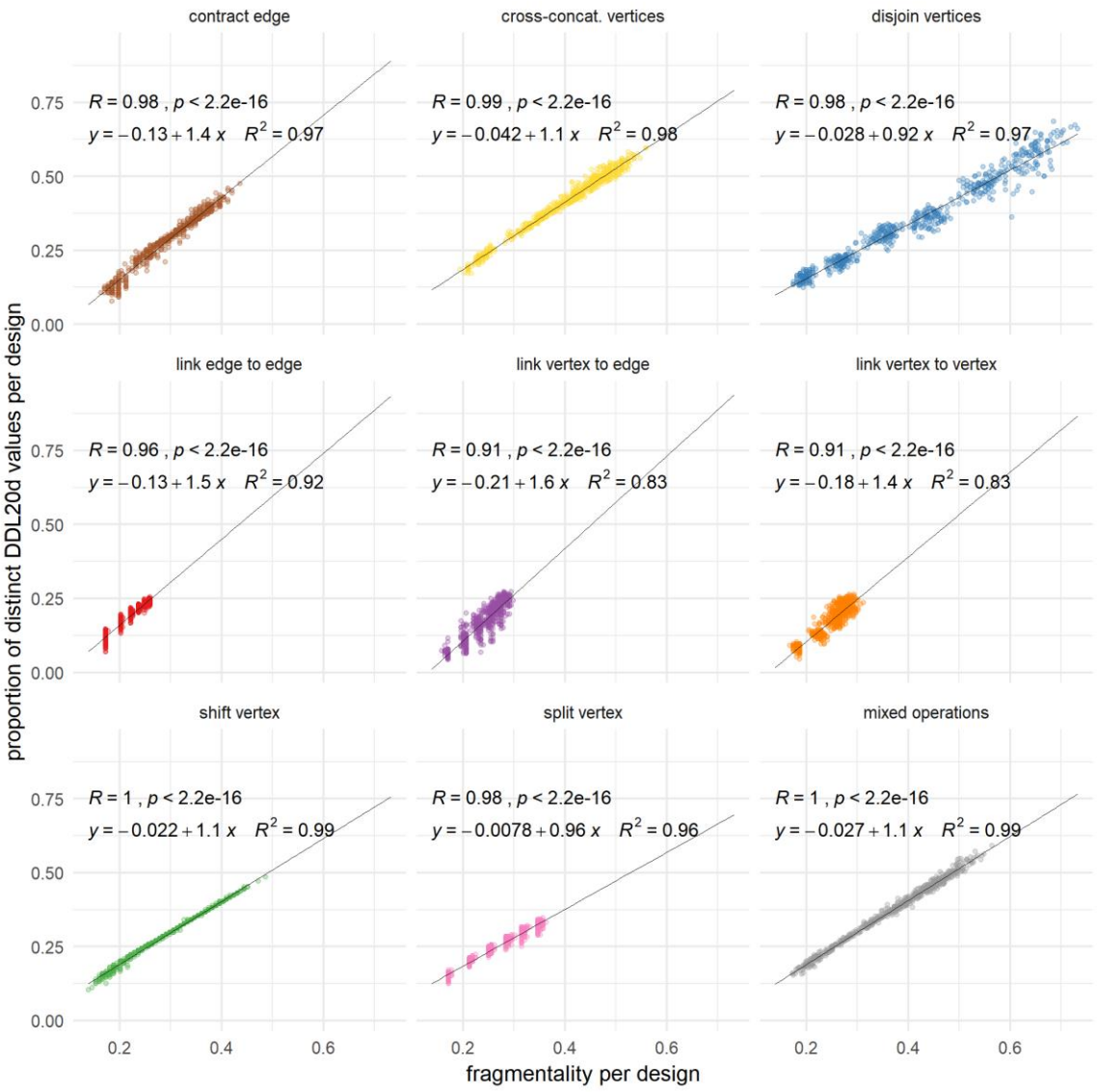

Figure 230. Scatterplots showing the relationship between proportion of distinct DDL20d values per design and fragmentality per design. Black lines are linear least squares regression fits to data points. In each plot, the regression line is extended to span the full range of the plot for better visibility.

This finding is important in two ways. First, it revealed and quantified the relationship between the irregularity associated with a design (as defined by the geometric alignment of streets) and the diversity associated with a design (as defined by the ratio of unique syntactic conditions). Second, it shows not only that there is indeed a

relationship between the two but that the relationship holds remarkably well across designs generated by different types of operations. In other words, no matter which type of operation is applied, given the same amount of increase in the fragmentality of a design, we would expect a consistent amount of increase in the proportion of distinct DDL20d values in that design.

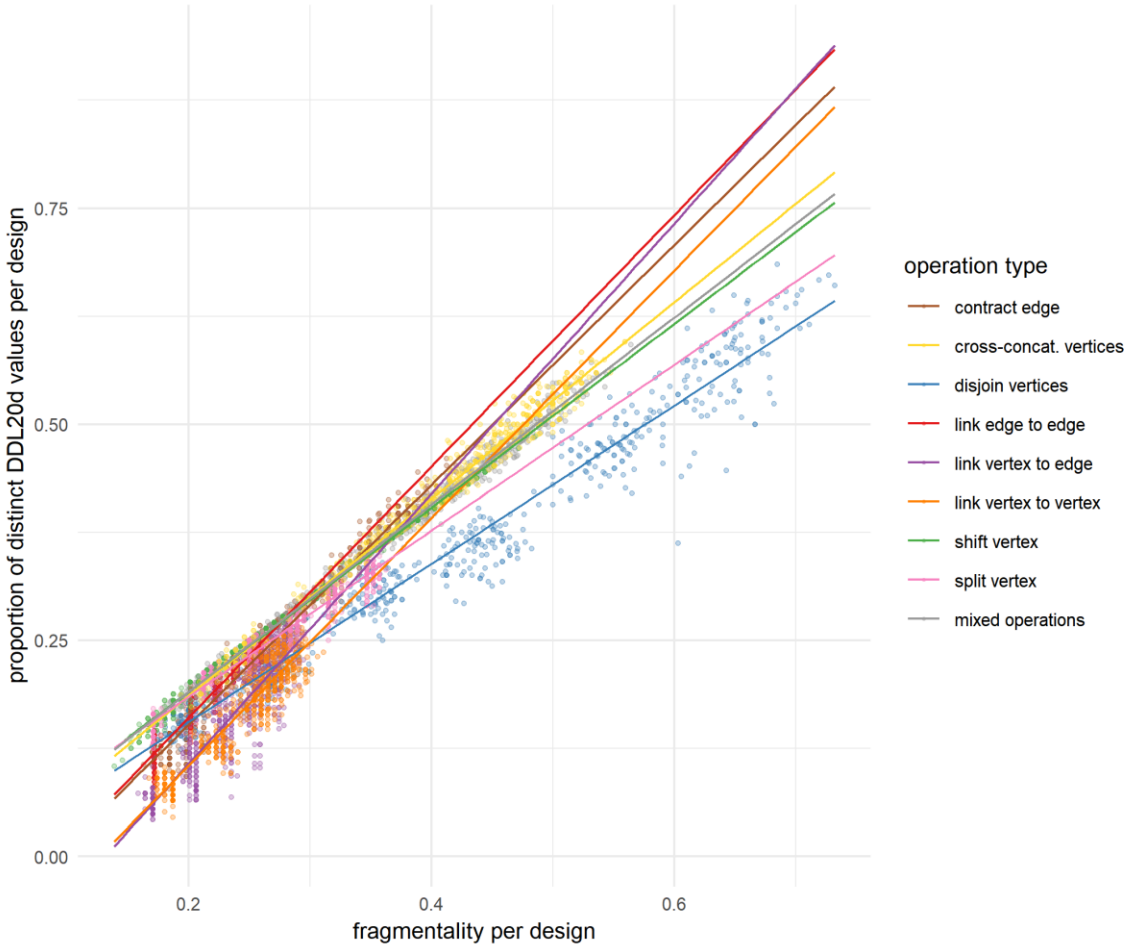

Figure 231. A scatterplot showing the relationship between proportion of distinct DDL20d values per design and fragmentality per design for each group (operation type). Linear regression lines are extended to span the full range of the plot for better visibility.

However, based on comparison of the group means across different measures (which we have presented in the first half of this chapter), we know that the increased fragmentality is also associated with changes in the other aspects of the design. Unlike the finding we have seen here, the relationship between the fragmentality and the other measures most of the time is not consistent across all designs, and it does seem to depend on the specific type of operation used. In other words, applying different operations to achieve the same amount of increase in fragmentality and diversity (in terms of the proportion of unique DDL values) could change the other aspects of the design in very different ways. Sometimes, the incurred changes in the other measures can be interpreted as certain costs associated with a unit increase in the fragmentality and diversity of the design. The following analyses show how the standard deviation of DDL values per design, the mean DDL per design, and the mean SAPR per design vary with the fragmentality per design.

10.2.3 Length-weighted standard deviation of DDL20d per design vs. fragmentality per design

For each group of designs, a Pearson correlation coefficient is computed to assess the relationship between the length-weighted standard deviation of DDL20d per design and the fragmentality per design. In almost all cases, there is a strong positive correlation between the two variables (Figure 232). However, the slopes of the regression lines are very different from each other (Figure 233). The finding suggests that, given the same amounts of increase in the fragmentality of a design (which further implies very similar amounts of increase in the proportion of DDL20d values), the associated amounts of increase in the standard deviation of DDL20d values are different for different groups of

designs, depending on the specific type of operation used to generate the design. For example, for the designs generated by linking edge to edge, linear regression analysis shows that for one unit increase in the fragmentality per design, the standard deviation of DDL20d is expected to be higher on average by 2.3. By contrast, for the designs generated by contracting edge, linear regression analysis shows that for one unit increase in the fragmentality per design, the standard deviation of DDL20d is only expected to be higher on average by 0.29.

Since fragmentality per design is a good proxy for the proportion of DDL20d values per design, the finding above, in a sense, suggests that varying the *amount* of differentiation (as measured by the proportion of distinct DDL20d values in a design) is not the same as varying the *degree* of differentiation (as measured by the standard deviation of DDL20d per design). To increase the proportion of DDL20d values by the same amount, we tend to increase the standard deviation of DDL20d values a lot more by linking edge to edge or disjoining vertices than, say, by contracting edge or cross-concatenating vertices.

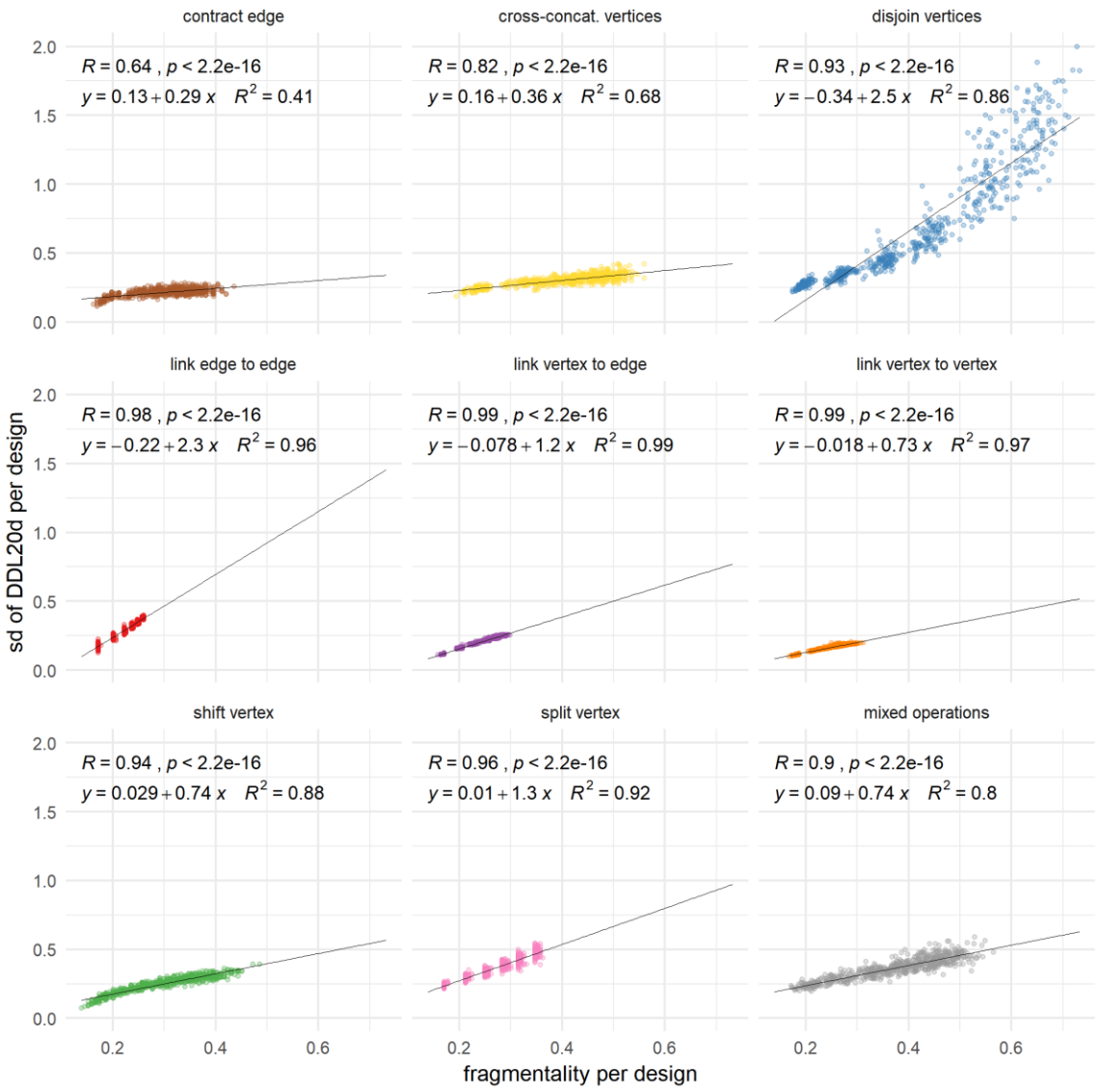

Figure 232. Scatterplots showing the relationship between length-weighted standard deviation of DDL20d per design and fragmentality per design. Black lines are linear least squares regression fits to data points. In each plot, the regression line is extended to span the full range of the plot for better visibility.

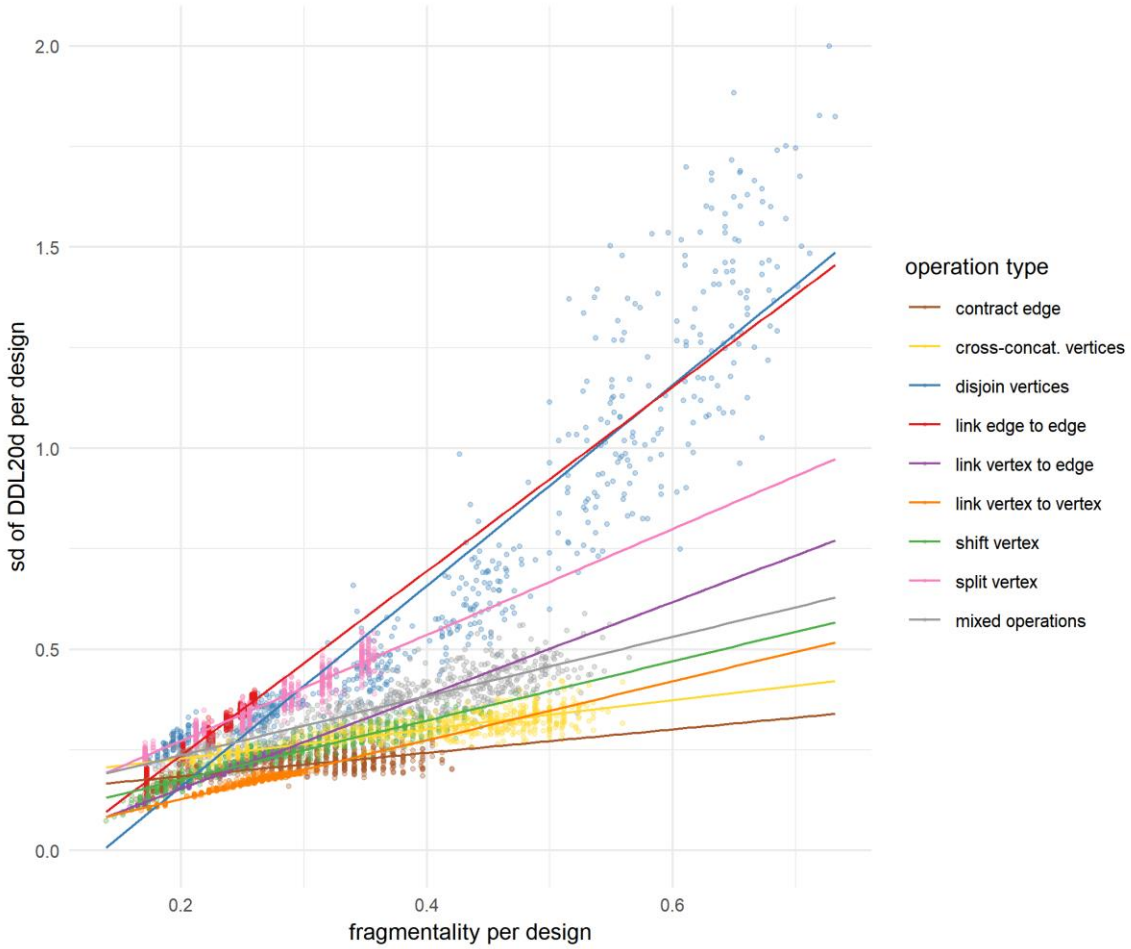

Figure 233. A scatterplot showing the relationship between length-weighted standard deviation of DDL20d per design and fragmentality per design for each group (operation type). Linear regression lines are extended to span the full range of the plot for better visibility.

10.2.4 Mean DDL20d per design vs. fragmentality per design

For each group of designs, a Pearson correlation coefficient is computed to assess the relationship between the mean DDL20d per design and the fragmentality per design. In almost all cases, there is a strong positive correlation between the two variables (Figure 234). However, the slopes of the regression lines are very different from each other (Figure 235). For the designs generated by splitting vertex, linear regression

analysis shows that for one unit increase in the fragmentality per design, the mean DDL20d is expected to be higher on average by 6.3. By contrast, for the designs generated by cross-concatenating vertices, linear regression analysis shows that for one unit increase in the fragmentality per design, the mean DDL20d is only expected to be higher on average by 0.87.

The increase of mean DDL suggests that, on average, more direction changes or turns need to be taken to move from one place to another place in the street network. This is usually undesirable because as the paths in the street network involve more turns, a greater cognitive effort is required in order to understand and navigate through the street network. Thus, we may consider the increase of the mean DDL as a kind of cost that we want to minimize in the design of a street network. The finding here suggests the different potentials associated with the different types of operations to minimize the cost of directional distance while diversifying the syntactic conditions at the same time. As we can see, to increase the diversity of syntactic conditions by the same amount (i.e., the same amount of increase in the proportion of distinct DDL20d values per design), we increase the mean DDL20d much more by splitting vertex or disjoining vertices than by cross-concatenating vertices.

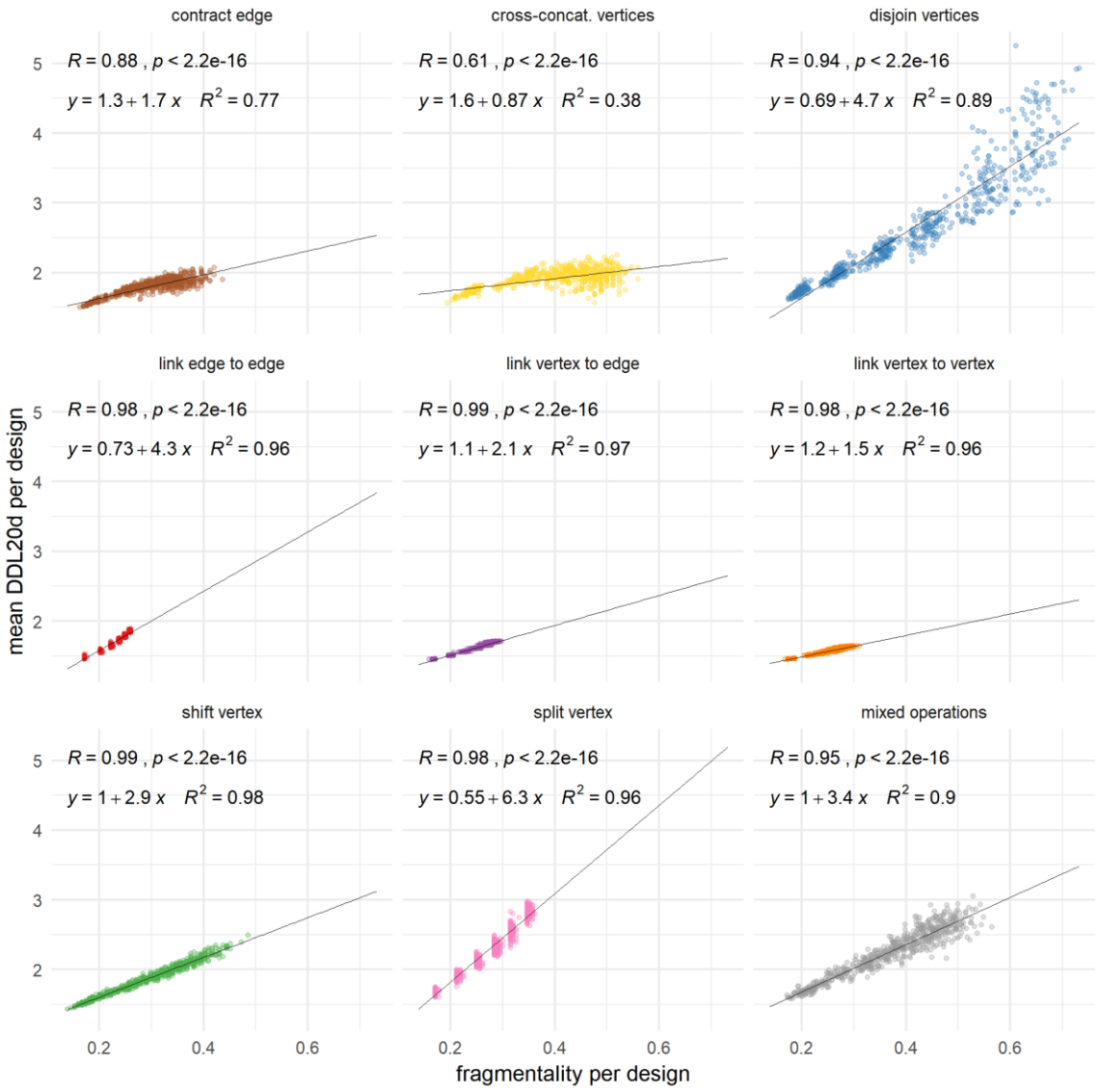

Figure 234. Scatterplots showing the relationship between mean DDL20d per design and fragmentality per design. Black lines are linear least squares regression fits to data points. In each plot, the regression line is extended to span the full range of the plot for better visibility.

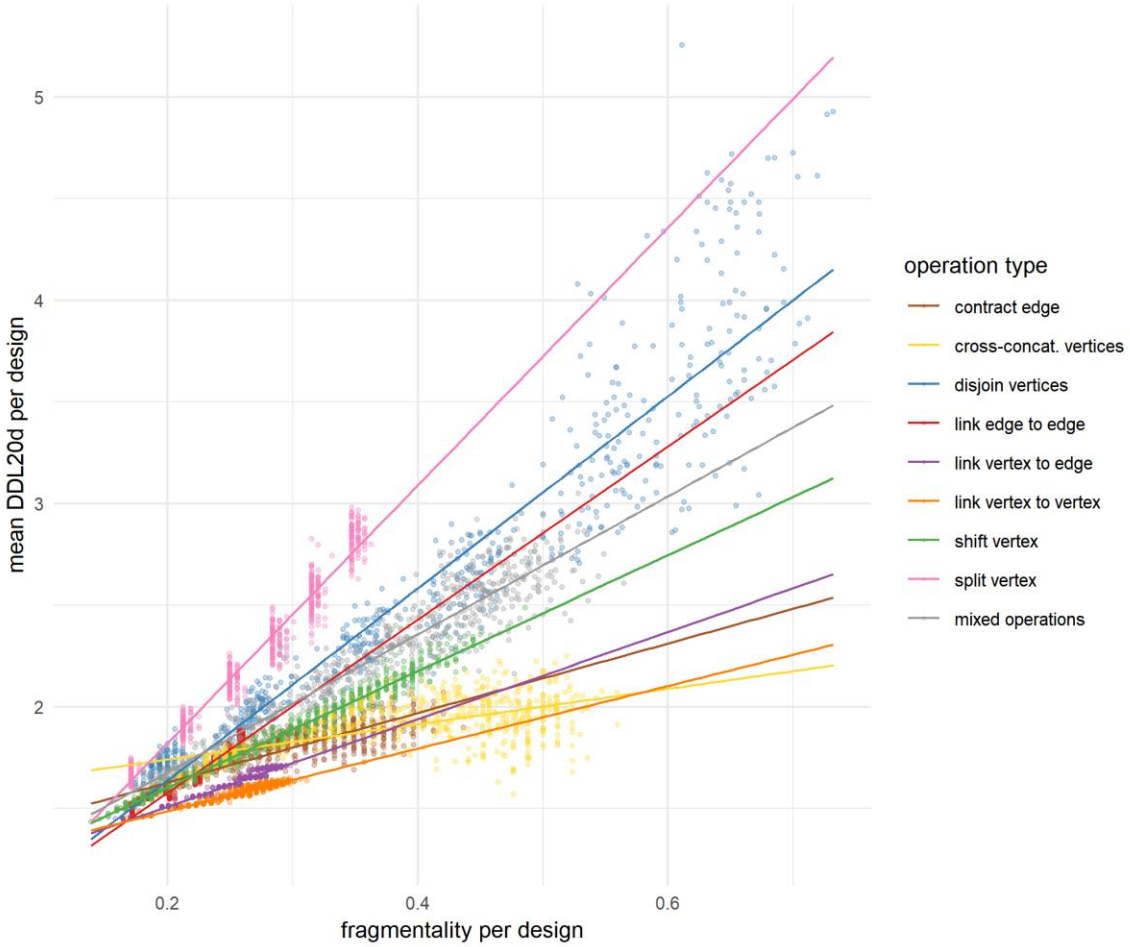

Figure 235. A scatterplot showing the relationship between mean DDL20d per design and fragmentality per design for each group (operation type). Linear regression lines are extended to span the full range of the plot for better visibility.

10.2.5 Mean SAPR per design vs. fragmentality per design

For each group of designs, a Pearson correlation coefficient is computed to assess the relationship between the mean standardized area-perimeter ratio (SAPR) per design and the fragmentality per design. In all cases, there is a strong negative correlation between the two variables (Figure 236). However, the slopes of the regression lines are very different from each other (Figure 237). For example, for the designs generated by

contracting edges, linear regression analysis shows that for one unit increase in the fragmentality per design, the mean SAPR is expected to be reduced on average by 0.54. By contrast, for the designs generated by shifting vertex, for one unit increase in the fragmentality per design, the mean SAPR is expected to be reduced on average by 0.12, which is a much less decrease.

The mean SAPR per design indicates the average compactness of block shape in a design, and a decrease in the mean SAPR per design suggests that, on average, the block shape in the design becomes less compact (and more different from a perfect square). Although variation in block sizes and shapes can often add visual interest to an urban area, very oddly shaped blocks and parcels require specialized designs which can take significantly more design effort. The finding here shows that, to increase the diversity of syntactic conditions by the same amount (i.e., the same amount of increase in the proportion of distinct DDL20d values per design), we decrease the mean SAPR much more by contracting edges or disjoining vertices than by shifting vertex or splitting vertex.

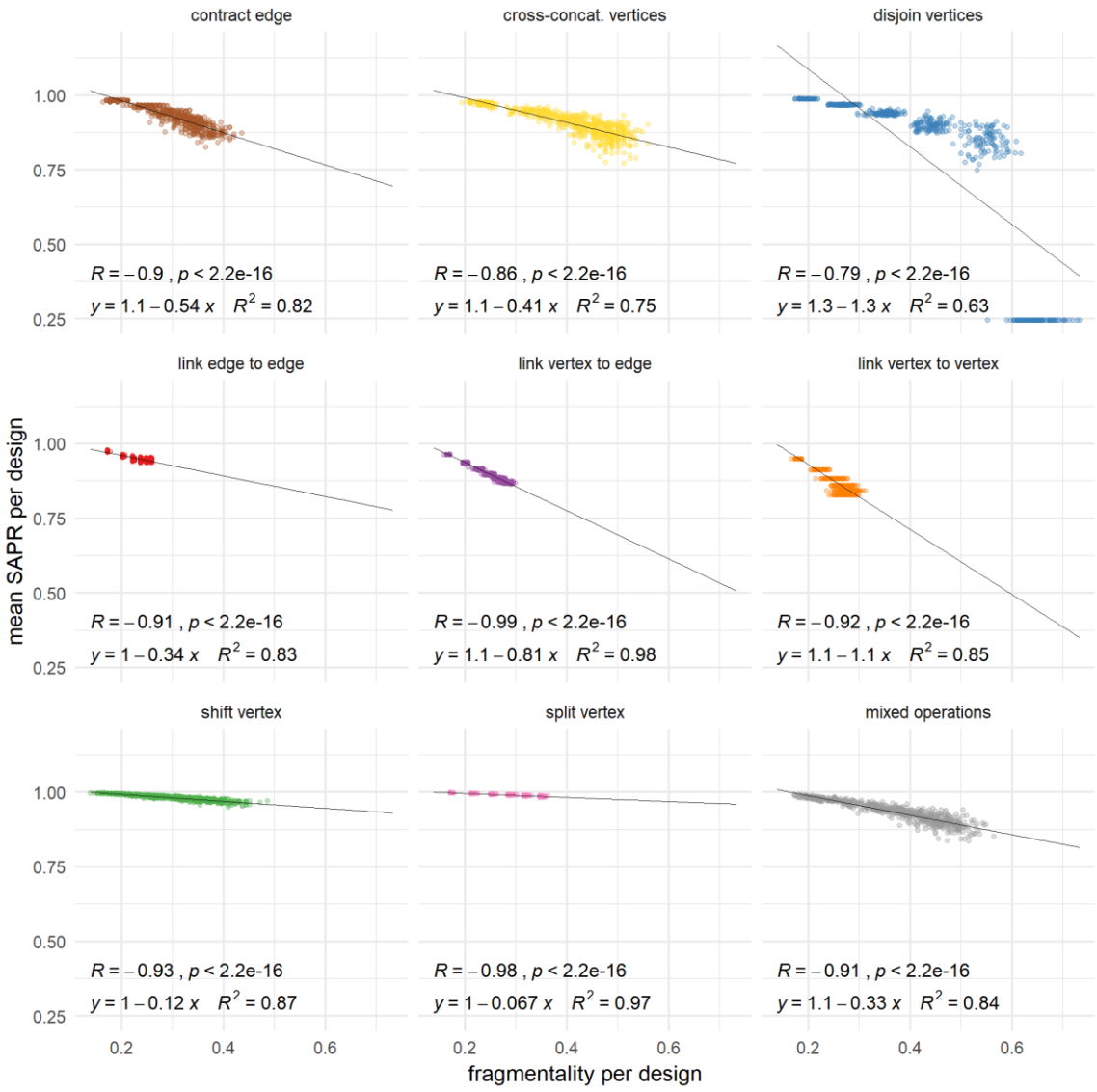

Figure 236. Scatterplots showing the relationship between mean SAPR per design and fragmentality per design. Black lines are linear least squares regression fits to data points. In each plot, the regression line is extended to span the full range of the plot for better visibility.

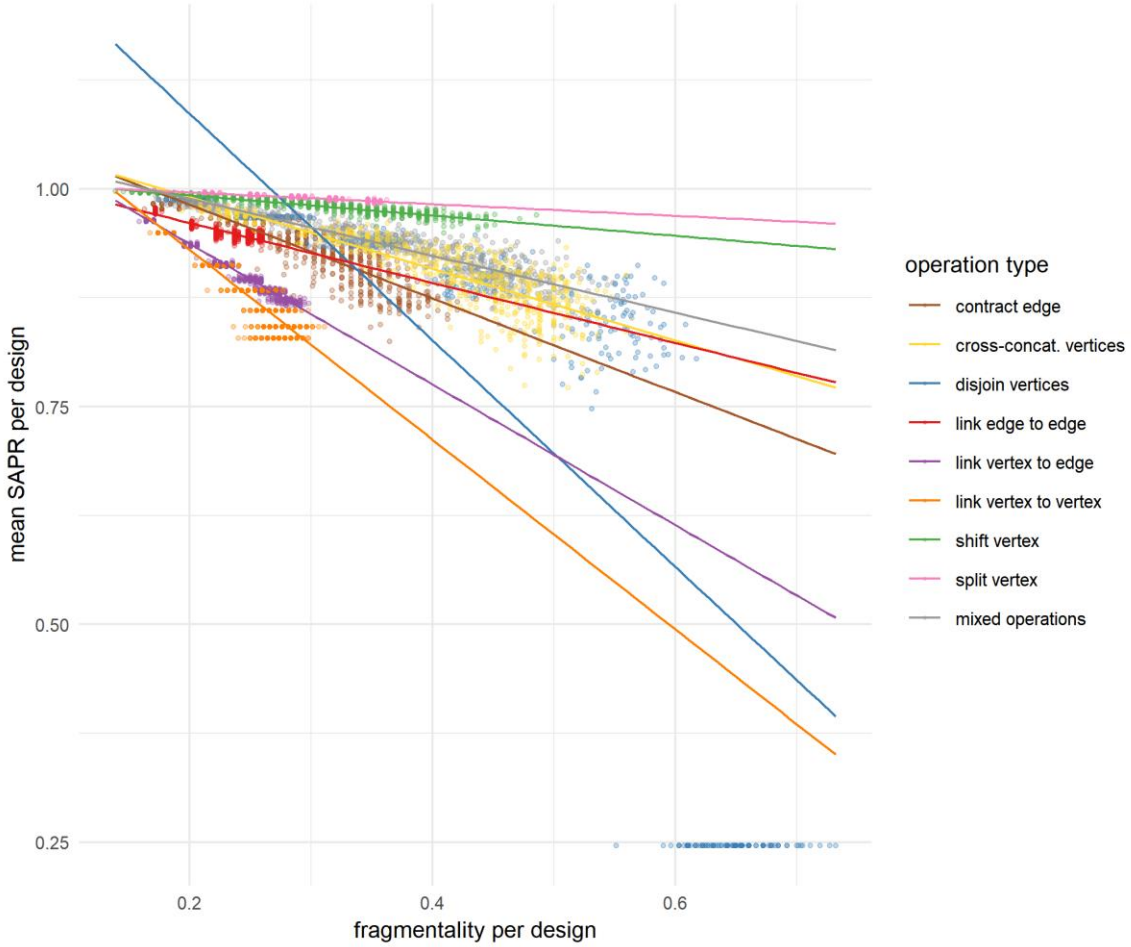

Figure 237. A scatterplot showing the relationship between mean SAPR per design and fragmentality per design for each group (operation type). Linear regression lines are extended to span the full range of the plot for better visibility.

As a square-grid design gets deformed, syntactic differentiation is created inside the design. If the degree of deformation is measured by the fragmentality per design, the above analysis shows that as a design is deformed by the same degree—no matter caused by which type of operation—a consistent increase in the proportion of distinct syntactic conditions is expected. However, that comes at various costs. Some costs are related to the increased directional distance, which may imply a greater cognitive effort to understand the environment. Some costs are related to the frequent occurrence of oddly

shaped blocks, which may pose a challenge for designers to incorporate them into a holistic design scheme. The kind of analysis conducted here helps us understand which type of operation can enrich the syntactic conditions in a design while minimizing unwanted costs.

In a broad sense, the above analysis fits into the research tradition of linking generative principles to syntactic outcomes started by Hillier and Hanson (Hillier, 1996b, 2002; Hillier & Hanson, 1984). However, with the exceptions of the chapter "Is architecture an ars combinatoria?" in *Space is the Machine* (Hillier, 1996b) and our recent paper on the superblock designs (Feng & Peponis, 2018), there has been little concerted effort to link generative syntactic principles to an analysis of the parametric variation of universes of designs. The work presented in this chapter follows that tradition and advances existing understandings by studying a much larger set of generative rules/principles and linking them to various aspects of designs.

CHAPTER 11
QUERY DESIGN UNIVERSES TO PROPOSE DESIGN TYPOLOGIES

11.1 Design Query

The 5400 superblock designs generated by the set of rules defined in previous chapters create a universe of designs which provides a basis for a systematic study of different properties of designs—either already well-known (closed-ended search) or emerging (open-ended search) during the process. The simplest kind of query is performed by sorting designs based on particular measures of interest. For example, we may be interested in finding the design with the lowest mean DDL, or the design with the most diverse syntactic conditions. However, there are other design properties which cannot be easily quantified. For example, if we would like to find the designs in which all the integrated streets (i.e. streets with relatively low DDL values) are located inside the boundary of the superblock and run parallel to each other, not only do we need to know the DDL value assumed by each street, but we also need to know the spatial relations between the streets.

In the field of space syntax, studies have shown that the location of and the relationship between the most integrated streets are important properties of a street network and suggest different perceptual, cognitive, and functional affordances of space. In space syntax, the set of the most integrated spaces of a spatial system is typically referred to as the *integration core* of the system (Hillier, 1996b; Hillier & Hanson, 1984). By convention, in an urban street system, the integration core consists of a given proportion, for example the 10 percent of the total number of streets (represented as axial

lines in the axial map) which take the highest integration values (Peponis et al., 1989). The form of the integration core can be studied at different spatial scales. By studying the syntactic structures of more than 50 cities across the world, Hillier (2002) found that those cities, in spite of their different cultural origins, share a similar pattern of integration core which resembles a "deformed wheel"—that is, a hub formed by the concentration of highly integrated streets around the heart of a city, spokes formed by long radial streets extending in all major directions, connecting the center and the periphery of a city, and a rim formed by the integrated streets running along the edges of a city. Similar integration core patterns can also be found in small and large towns—such as the French towns of Apt and Perpignan—and in urban areas down to the scale of a superblock—such as those observed in the Gangnam District of Seoul, Korea (Hillier et al., 1983; Peponis et al., 2015; Peponis et al., 2016).

In the next section, we explore and study a typology of superblock designs based on the integration core patterns observed in the design space created.

11.2 Types of Integration Cores

11.2.1 Definition of an integration core

As mentioned above, by convention, the integration core is identified by selecting the 10 percent (or 5 percent for big cities) of the total number of axial lines in the axial map which take the highest values of integration. Although in many cases informative results can be obtained by following this convention, there is no strong theoretical support for the percentage conventionally used. A serious defect in this method is due to its ignorance of the distribution of the data set—we do not know how different the

integration core is from the rest of the system. To address this issue, we define the integration core by selecting the continuity lines whose DDL values are a certain number of standard deviations below the mean DDL of the whole system. The size of an integration core is measured by the total length of the continuity lines that comprise the integration core, without double counting the overlapped portions. Integration cores can be parametrically defined: we can relax the criterion of what makes up an integration core by specifying a smaller number of standard deviations below the mean DDL or, conversely, be more strict about it by specifying a larger number of standard deviations below the mean DDL. But more importantly, the integration core defined as such is now sensitive to the distribution of the data set: the sizes of the integration cores may grow at distinct paces as we parametrically adjust the mentioned criterion, depending on the distribution of DDL values within a design. We also define the integration core based on continuity lines instead of segments to acknowledge linear extensions of streets.

As shown in Figure 238, the size of the integration core generally becomes larger as the criterion of what makes up the integration core is gradually relaxed. However, designs generated by different rules and operations exhibit distinct trajectories of growth.

For the designs generated by the operations of linking edge to edge, linking vertex to edge, and linking vertex to vertex, the growth of the integration core is almost invisible initially—there are barely any streets meet the strict selection criterion of the integration core. After we relax the criterion by 0.3 or 0.4 standard deviations, however, there comes a sudden and dramatic growth of the integration core—from almost nothing to a substantial part of the entire street network. Then, as we keep relaxing the criterion, the size of the integration core quickly becomes stable again. The “step-like” increase

reflects the distribution of DDL20d values for the designs generated by those operations: a cluster of low DDL20 values assigned to the streets that comprise the main grid, and a cluster of high DDL20d values assigned to the streets nested inside the square blocks. The two clusters of DDL values are very different from each other, but within each cluster, the values are not very differentiated. Therefore, the low DDL values are easily missed out or captured altogether.

By contrast, the growth trajectories of the size of the integration core for the designs generated by the other kinds of operations exhibit "ramp-like" patterns of increase, although still slightly varying from each other in the overall rate of change. This "ramp-like" pattern of increase suggests that, for those designs, the group of below-average DDL values are still sufficiently differentiated within themselves.

The absolute sizes of the integration core identified based on different criteria are also interesting. For example, excluding the three operations of linking, the median integration core size for the designs generated by the operation of disjoining vertices is consistently the lowest among all (Figure 238B). We have also relativized the integration core size by the total street length per design. Overall, the growth trajectories indicating the relativized core size look similar to those indicating the absolute core size (Figure 239A). Nonetheless, there are a few notable differences. For example, without considering the three linking operations, designs generated by cross-concatenating vertices now tend to have the highest relativized core size. Moreover, although the designs generated by disjoining vertices tend to have the lowest absolute core sizes, the sizes, after relativization, are now comparable to the designs generated by the other operations (Figure 239B).

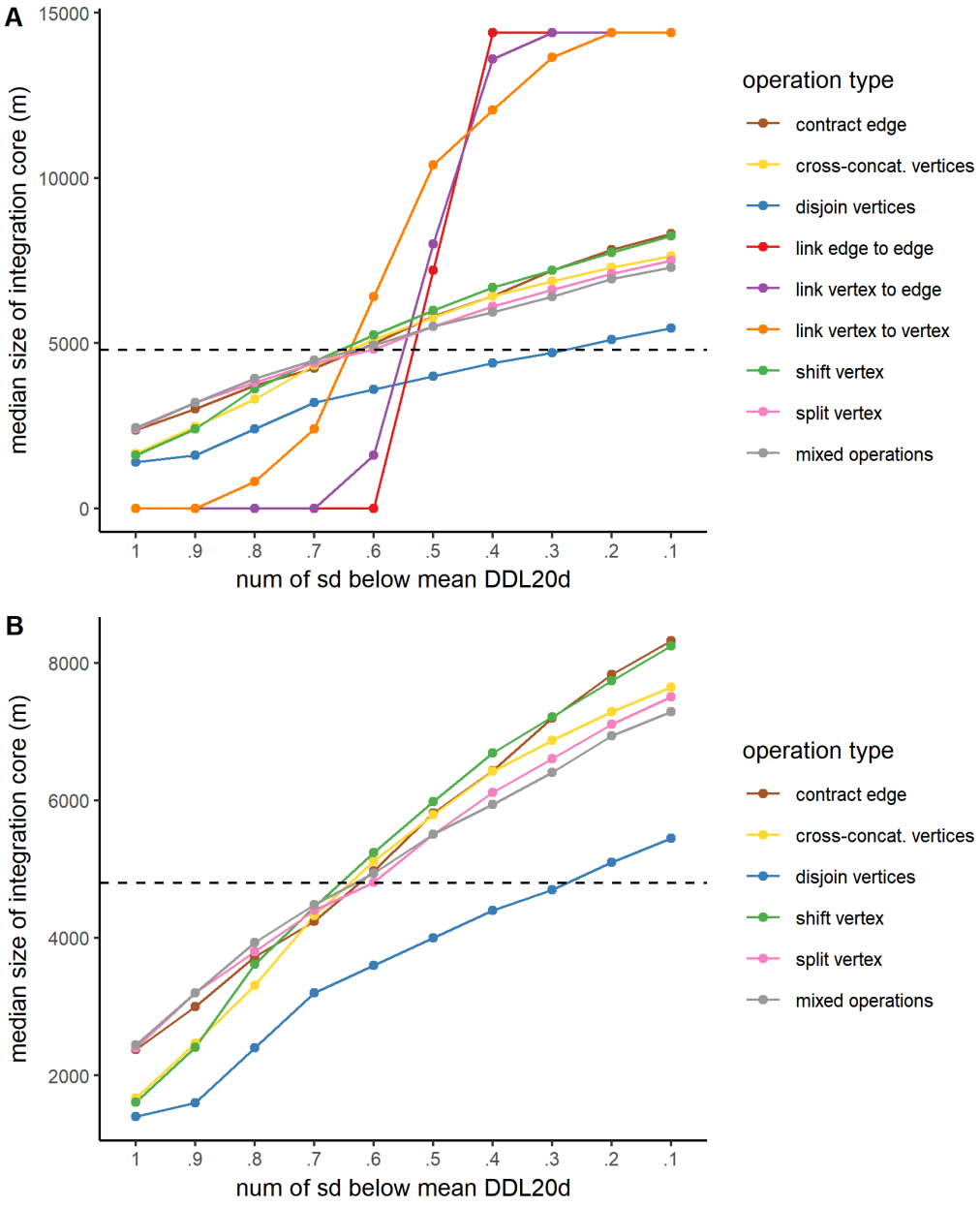

Figure 238. Line graphs showing the change in the median size of the integration core (measured by length) as the criterion of what makes up the integration core is gradually relaxed. A: All types of operations are included. B: All types of operations are included except the operations of linking edge to edge, linking vertex to edge, and linking vertex to vertex. In both subfigures, the black dashed line indicates the minimal length of streets necessary to comprise a prototypal deformed-wheel-type integration core (4800 m).

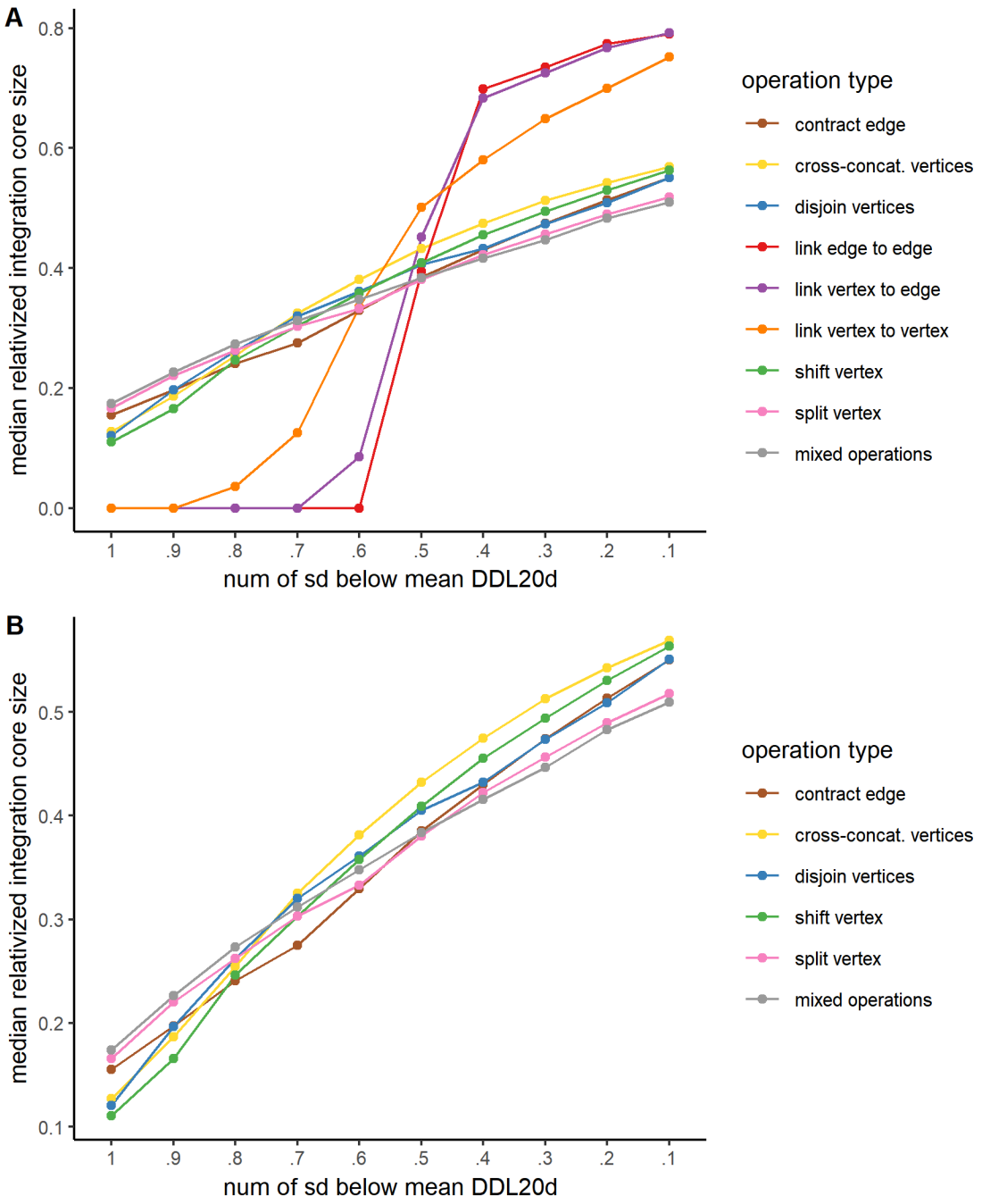

Figure 239. Line graphs showing the change in the median relativized integration core size as the criterion of what makes up the integration core is gradually relaxed. A: All types of operations are included. B: All types of operations are included except the operations of linking edge to edge, linking vertex to edge, and linking vertex to vertex.

11.2.2 Schematic expressions and algorithmic definitions of different types of integration cores

Now that we have given a definition of an integration core, we are ready to present three types of integration cores observed in the design space created: (a) radial-type cores, (b) spine-type cores, and (c) rim-type cores.

Before we explain the different types of integration cores, we first put the continuity lines into several different categories. As shown in Figure 240, assuming that the superblock is aligned with the x- and y-axes, there are generally three ways for a continuity line to traverse a superblock: (a) diagonally, (b) horizontally, or (c) vertically. A diagonal traversing continuity line starts at one corner of the superblock and ends at the opposite corner of the superblock. A horizontal traversing continuity line traverses the superblock horizontally without touching the opposite corners of the superblock. (Therefore, diagonal traversing continuity lines are not considered to be horizontal ones.) Similarly, a vertical traversing continuity line traverses the superblock vertically without touching the opposite corners of the superblock.

In the hypothetical superblock design shown in Figure 240, the continuity line *D-S-X-W-L* is the only diagonal traversing continuity line in the design. The horizontal traversing continuity lines include the continuity lines *A-P-N-M-L*, *C-S-T-U-H*, and *D-E-F-G*, and the vertical traversing continuity lines include the continuity lines *A-B-C-D*, *P-Q-R-S-E*, *N-Y-X-T-F*, and *L-K-J-H-G*.

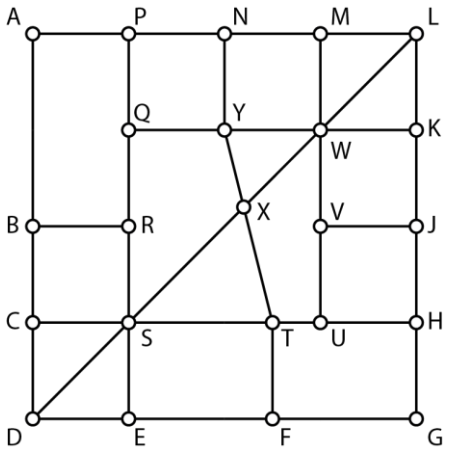

Figure 240. A hypothetical design showing different types of continuity lines.

For the horizontal and vertical traversing continuity lines, further differentiation can be made based on whether the continuity line lies on the border of the superblock acting as a bounding street or located inside the superblock. Since we never deform the boundary of the superblock according to all of our generative algorithms, every design in our design space has four traversing continuity lines together defining the boundary of the superblock, with two running horizontally and two vertically.

Radial-type integration cores

A radial-type integration core is made up of at least two continuity lines which traverse the superblock in opposite directions and intersect at a location near the geometric center of the superblock. Therefore, the minimum size of a radial-type integration core is the sum of the length of a horizontal traversing continuity line and that of a vertical traversing one—that is, 800 + 800 = 1600 m.

For a specific design, suppose that we have already identified its integration core which consists of a number of continuity lines. The algorithm used to check whether the design has a radial-type integration core is described as follows.

1. Consider the set of lines that are included in the integration core and select the continuity lines that traverse the superblock.
2. Label the selected traversing continuity lines with one of the following: (a) diagonally traversing, (b) horizontally traversing, and (c) vertically traversing. Based on the distinct labels, the traversing continuity lines selected in the first step fall into three disjoint sets.
3. Record any point of intersection between continuity lines coming from distinct sets derived in the above step. In addition, record any point of intersection between the diagonal traversing continuity lines (to detect the rare case that the radials are shaped by two diagonal traversing continuity lines that cross each other near the center).
4. Check if there are any points of intersection detected in the above step. For each point of intersection, compute its distance from the geometric center of the superblock. If at least one point of intersection is detected and its distance from the center of the superblock is not more than a parametrically set value, for example 200 m, then the design is considered to have a radial-type integration core.

Schematic expressions of the radial-type integration cores are illustrated in the top rows of Figure 241.

The deformed-wheel like integration core as mentioned before thus becomes a special case of the radial-type integration core where the continuity lines on the boundary of the superblock are also part of the integration core. We may call the integration cores with both the radials and the rims on all four sides of the superblock a *full-deformed-wheel-type* integration core. The three schematic expressions of the full-deformed-wheel-type integration core are shown in the rightmost column of the top rows of Figure 241.

Spine-type integration cores

A spine-type integration core consists of one or more traversing continuity lines inside the superblock without crossing each other. The minimum size of a spine-type integration core is the length of a horizontal or vertical traversing continuity line, which is 800 m.

For a specific design, suppose that we have already identified its integration core. The algorithm used to check whether the design has a spine-type integration core is described as follows.

1. In the integration core, select the traversing continuity lines that lie inside the superblock.
2. If no continuity lines have been selected in the first step, then the design does not have a spine-integration core; if there are more than one continuity lines selected in the first step and there is at least one point of intersection between those lines, then the design does not have a spine-type integration core; in all the other cases, the design is considered to have a spine-type integration core.

Schematic expressions of the spine-type integration cores are illustrated in the middle rows of Figure 241.

Rim-type integration cores

A rim-type integration core consists of only continuity lines on the boundary of the superblock. The minimum size of a rim-type integration core is the length of a traversing continuity line on the boundary of the superblock, which is 800 m.

For a specific design, suppose that we have already identified its integration core. The algorithm used to check whether the design has a rim-type integration core is described as follows.

1. In the integration core, select all the traversing continuity lines.
2. If any of the continuity lines selected in the above step lies inside the superblock, the design does not have a rim-type integration core. Otherwise, the design is considered to have a rim-type integration core.

Schematic expressions of the rim-type integration core are illustrated in the bottom row of Figure 241.

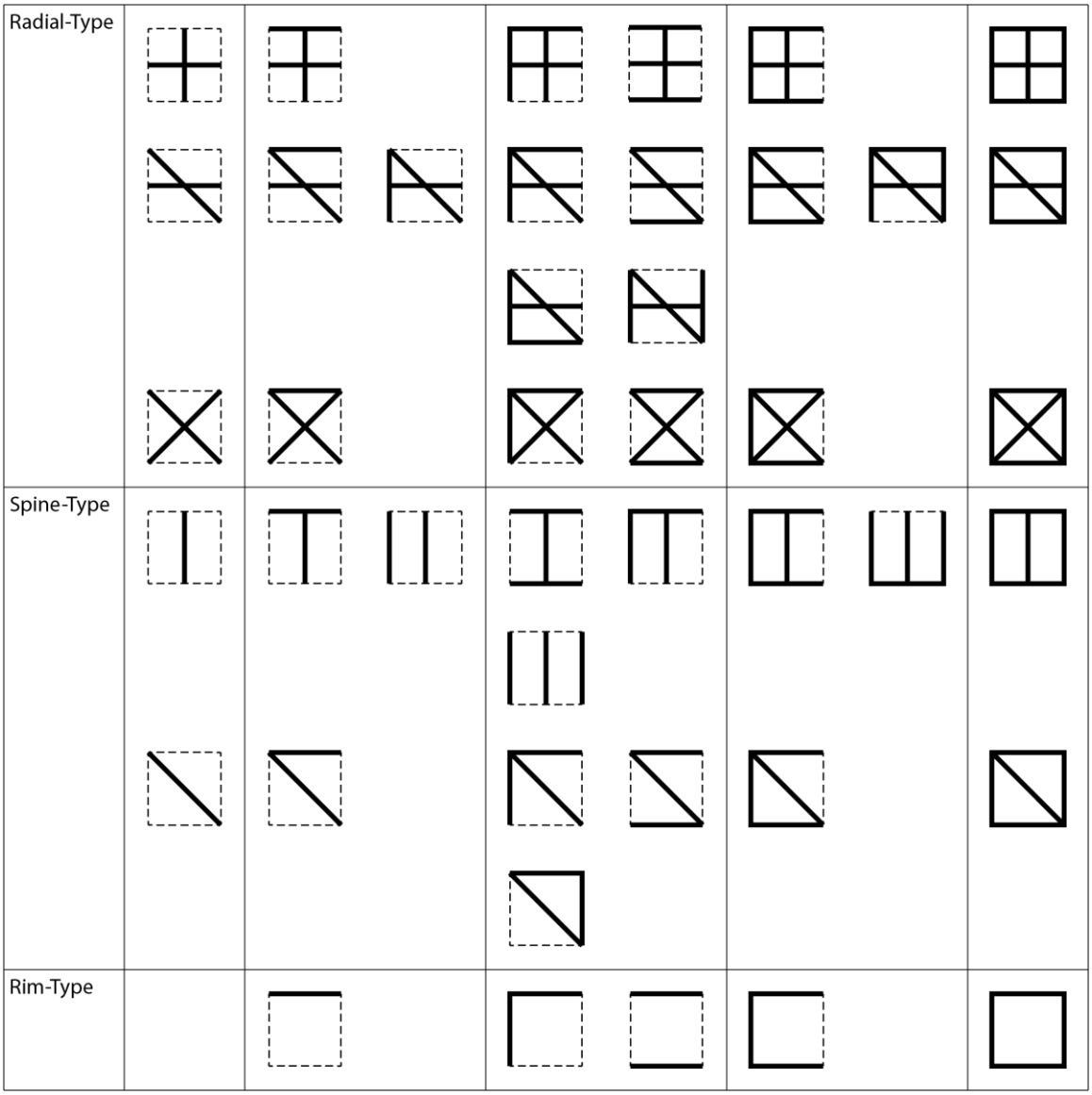

Figure 241. Schematic expressions of the different types of integration cores. In every row, the number of continuity lines lying on the boundary of the superblock (as part of the integration core) gradually increases from left to right.

11.2.3 Query results

A series of queries has been conducted to find designs with the different types of integration cores. In conducting the queries, we gradually relaxed the criterion of what makes up an integration core. According to the most strict criterion, the integration core consists of the continuity lines whose DDL20d values are at least one standard deviation below the mean DDL20d of the design. According to the most relaxed criterion, the integration core consists of the continuity lines whose DDL20d values are at least 0.1 standard deviations below the mean DDL20d of the design.

Designs with radial-type integration cores

As shown in Figure 242, the designs are separated into nine groups based on the different types of operations that have generated them. Overall, the number of designs with a radial-type integration core increases as the criterion of what makes up an integration core is gradually relaxed. However, the growth trajectories in the number of designs with a radial-type core are different for different groups of designs. For example, for the groups of designs generated by the operations of linking vertex to vertex, linking vertex to edge, and linking edge to edge, the rate of growth in the number of radial-type integration cores is much faster than the other groups (after we have relaxed the criterion by 0.1, 0.2, and 0.3 standard deviations of DDL20d, respectively). When the most relaxed criterion is used, *all* designs from those three groups are considered to have a radial-type integration core according to our query algorithm. Their growth trajectories reflect the sudden increases in the sheer size of the integration core (as before shown in Figure 238A) and the two-level street system as explained before.

By contrast, the growth trajectories for the other groups are more or less comparable, although still different from each other. As shown in Figure 242B, excluding the three operations of linking, the operation of shifting vertex always yields the highest number of designs that have a radial-type integration core. The growth trajectories for the groups associated with the operations of contracting edge, cross-concatenating vertices, and splitting edges are quite similar, squeezed between the trajectories associated with the operation of shifting vertex and the operation of disjoining vertices and mixed operations. The operation of disjoining vertices and the mixed operations consistently yields the lowest number of designs that have a radial-type integration core. In all cases of the six operations (excluding the three linking operations), the number of designs with the radial-type integration core yielded from each type of operation becomes more or less stabilized after we have relaxed the criterion by about 0.5 or 0.6 standard deviations of DDL20d per design.

Examples of designs with radial-type integration cores are shown in Figure 243. In generating the examples, we define the integration core as the set of continuity lines whose DDL20d values are at least 0.5 standard deviations below the mean DDL20d.

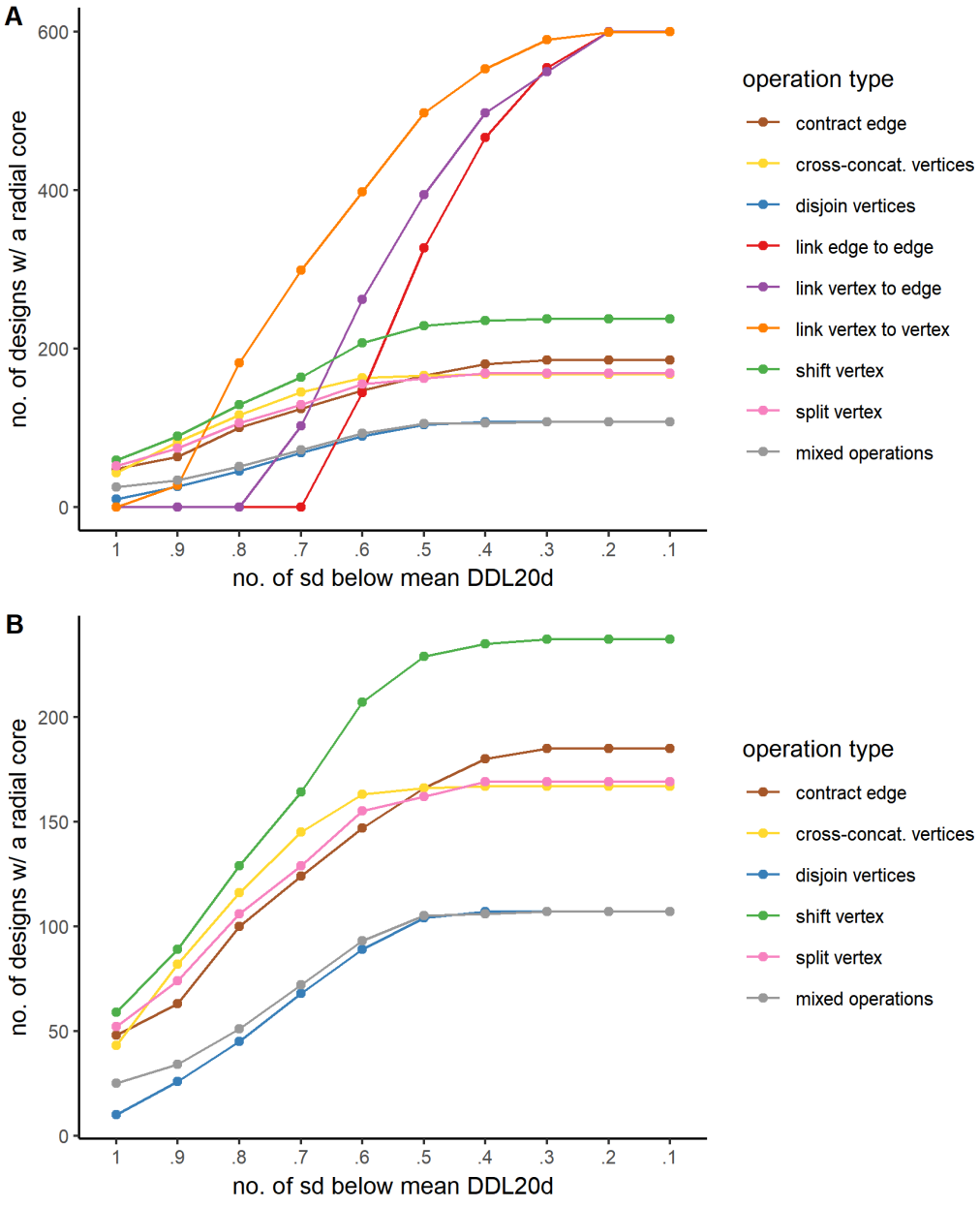

Figure 242. Line graphs showing the change in the number of designs that have a radial-type integration core as the criterion of what makes up the integration core is gradually relaxed. A: All types of operations are included. B: All types of operations are included except the operations of linking edge to edge, linking vertex to edge, and linking vertex to vertex.

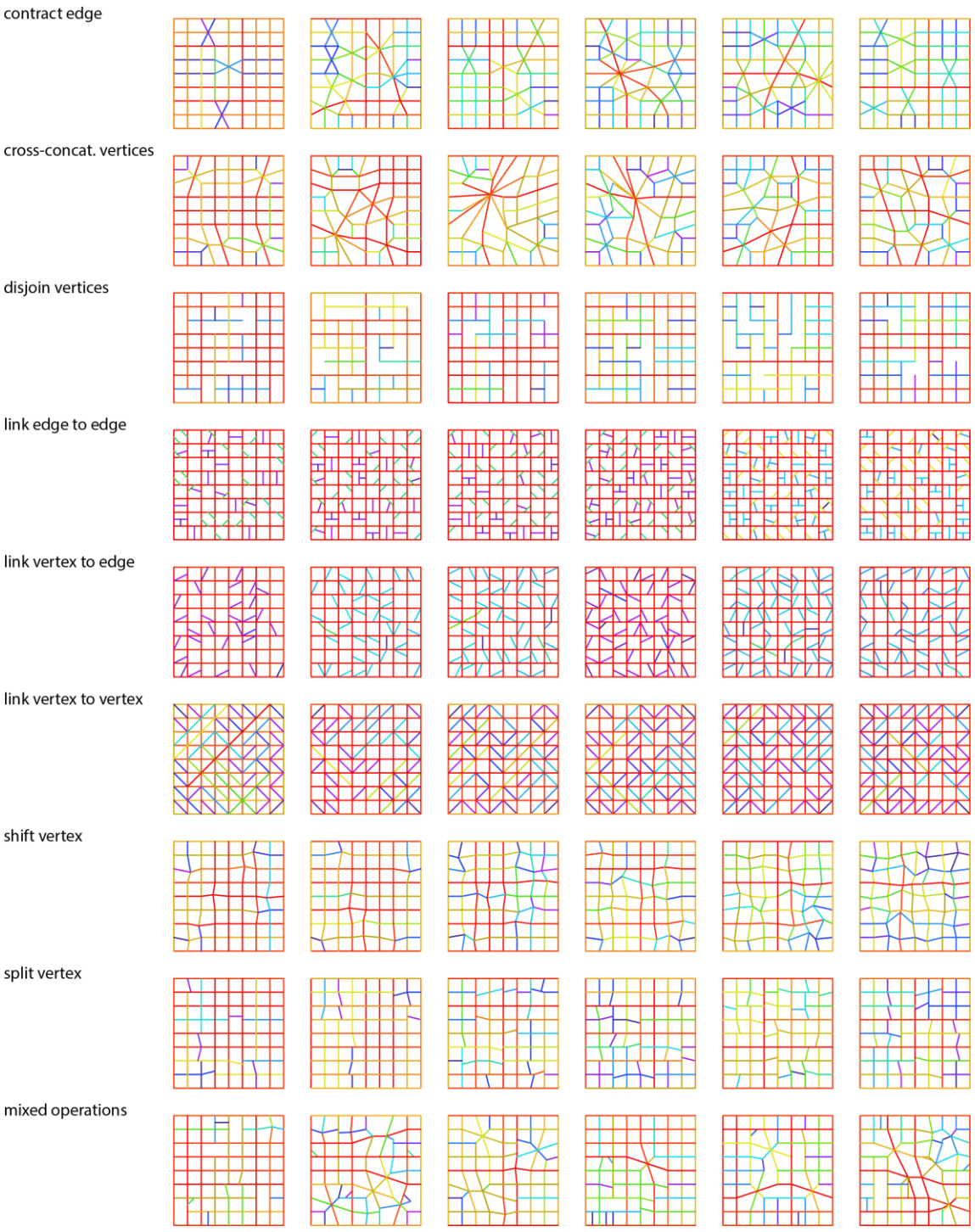

Figure 243. Examples of designs with radial-type integration cores selected from each group of designs.

Designs with full-deformed-wheel-type integration cores

As shown in Figure 244, the number of designs with a full-deformed-wheel-type integration core increases as the criterion of what makes up an integration core is gradually relaxed. The operations of linking vertex to vertex, linking vertex to edge, and linking edge to edge again stand out for the explosive growth in the number of designs that have a full-deformed-wheel-type integration core after we have relaxed the criterion of what makes up an integration core by 0.4 or 0.5 standard deviations. When the most relaxed criterion is adopted, almost all designs in those three groups have a full-deformed-wheel-type integration core.

By contrast, among the other groups, the number of designs with a full-deformed-wheel-type integration core grows at a much gentler pace. Most of the time, the operation of cross-concatenating vertices yields the highest number of designs that have a full-deformed-wheel-type integration core. It is interesting to see that the operations of shifting vertex and contracting edge, although yield more designs with radial-type integration cores, produce fewer designs with full-deformed-wheel-type integration cores as compared to the operation of cross-concatenating vertices after we have relaxed the criterion of what makes up an integration core by 0.4 standard deviations and more. In fact, the operation of contracting edge, along with the operation of disjoining vertices and the mixed operations, tend to produce the lowest number of designs that have a full-deformed-wheel-type integration core.

Examples of designs with full-deformed-wheel-type integration cores are shown in the rightmost column of Figure 243.

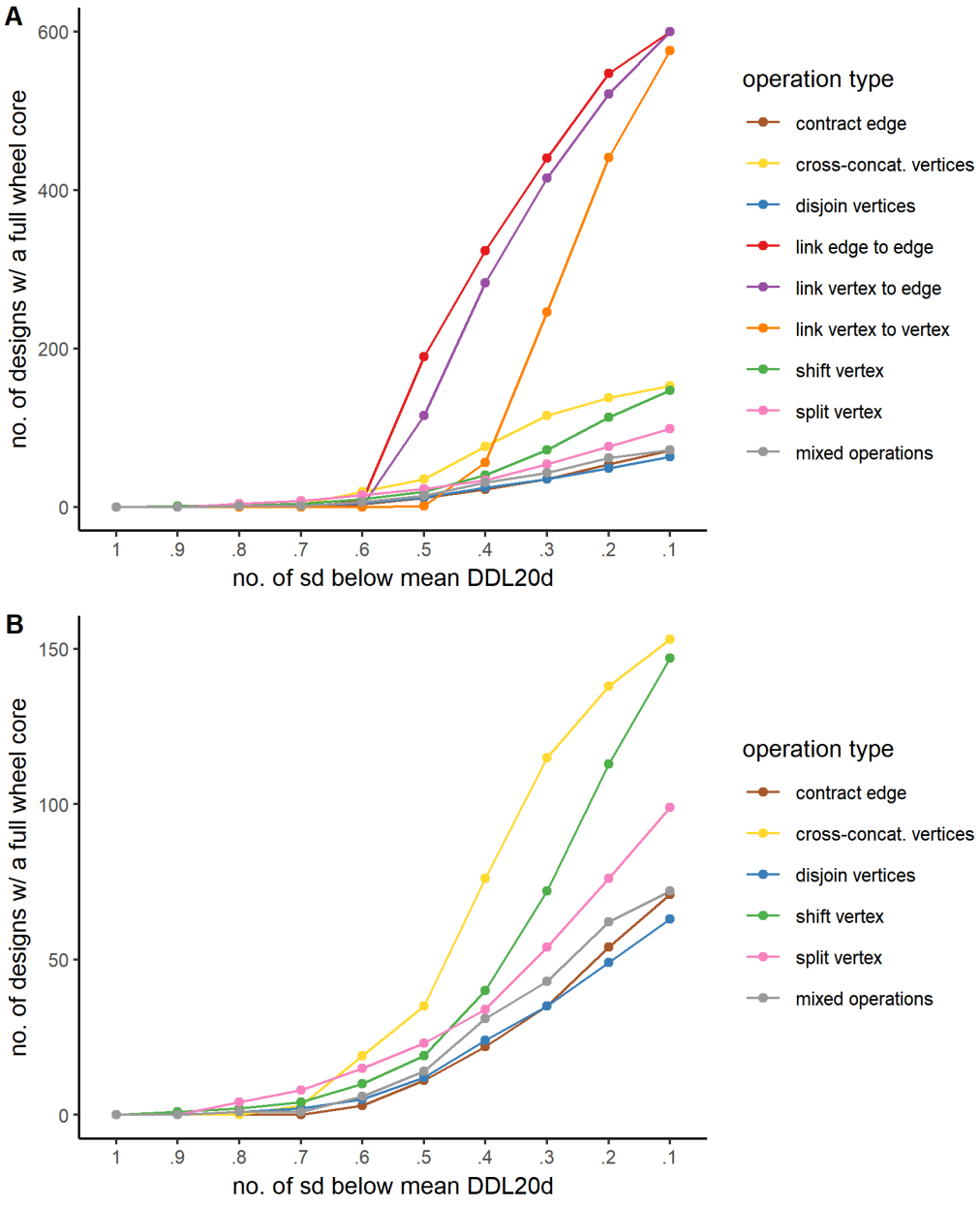

Figure 244. Line graphs showing the change in the number of designs that have a deformed-wheel-type integration core as the criterion of what makes up the integration core is gradually relaxed. A: All types of operations are included. B: All types of operations are included except the operations of linking edge to edge, linking vertex to edge, and linking vertex to vertex.

Designs with spine-type integration cores

As shown in Figure 245, as the criterion of what makes up an integration core is gradually relaxed, the number of designs with a spine-type integration core for the groups of designs generate by the operations of linking vertex to vertex, linking vertex to edge, and linking edge to edge increases first, then tends to stabilize for a short while, and then drops again and reaches zero in the end. The initial increase and the later drop are due to the two-level system and the configuration of the main grid. When strict criteria are adopted, few continuity lines are included in the integration core, thus there is little chance to form the "spines". As the criterion is relaxed a bit, some traversing continuity lines on the main grid of the superblock are included in the integration core, resulting in the initial increase of the number of designs with spine-type cores. However, the continuity lines on the main grid of the designs generated by the linking operations have similar DDL values and tend to be included altogether in the integration core. Since those continuity lines on the main grid cross each other at frequent intervals, there is little chance for them to stay as spine-type integration cores and not be converted into the radial-type cores. As a consequence, among those designs, the number of designs with a spine-type integration core decreases as the number of designs with a radial-type integration core increases.

By contrast, for the rest of the operations (except for the operation of cross-concatenating vertices), the number of designs with a spine-type integration core slightly increases at first, then gently drops, and then stabilizes. Generally, designs with spine-type integration cores are more likely to be produced by applying the operation of contracting edge or the operation of cross-concatenating vertices on a square grid. They

are least likely to be produced by applying the operation of disjoining vertices because there tend to be fewer traversing continuity lines inside the superblock design.

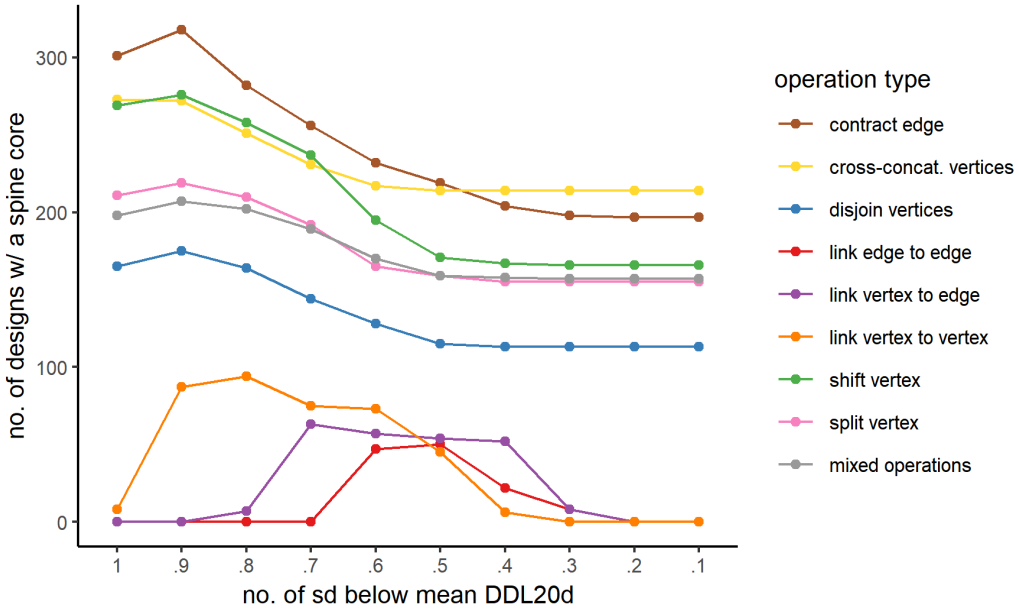

Figure 245. A line graph showing the change in the number of designs that have a spine-type integration core as the criterion of what makes up the integration core is gradually relaxed.

Examples of designs with a spine-type integration core are shown in Figure 246. In generating the examples, we define the integration core as the set of continuity lines whose DDL20d values are at least 0.5 standard deviations below the mean DDL20d.

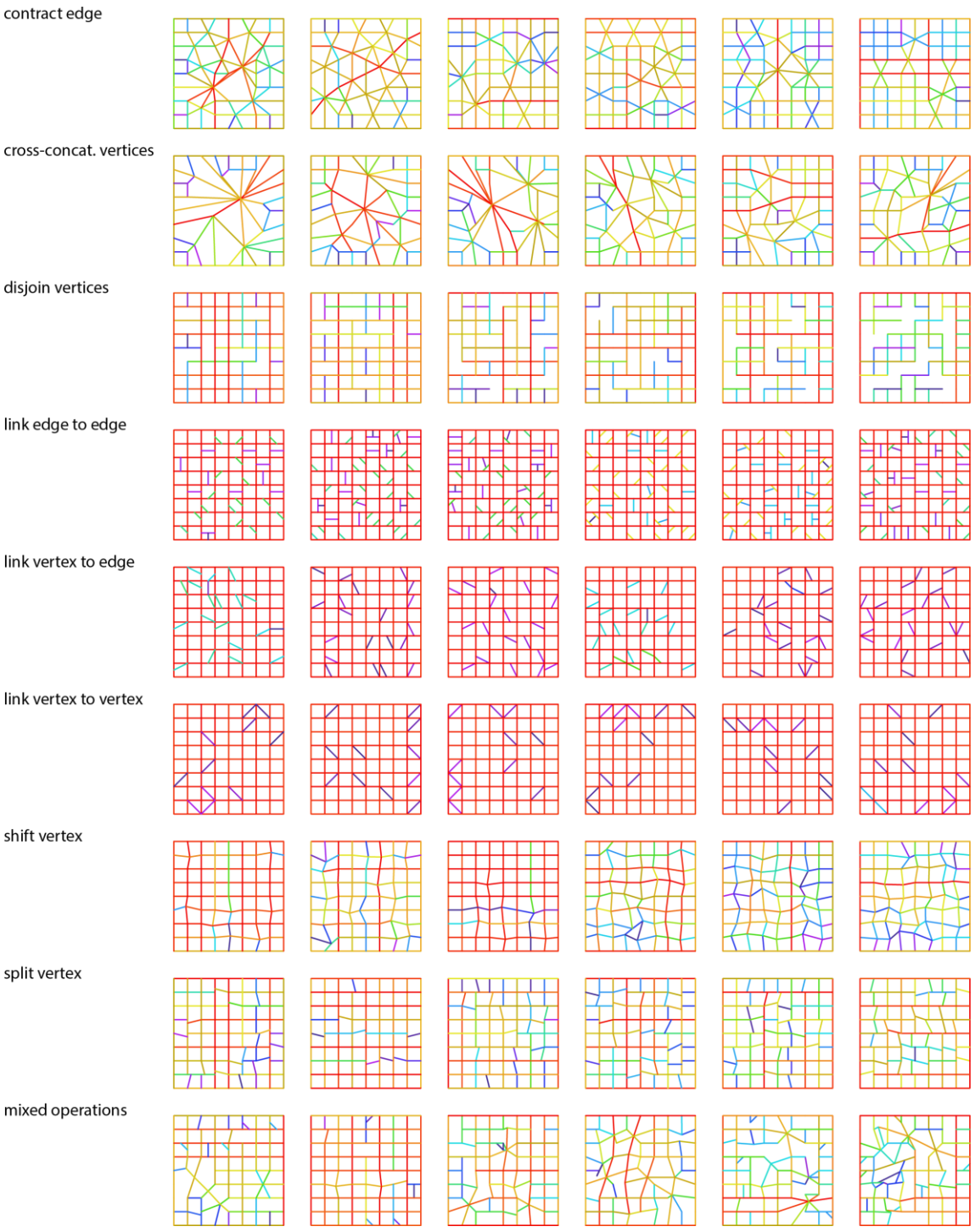

Figure 246. Examples of designs with spine-type integration cores selected from each group of designs.

Designs with rim-type integration cores

As shown in Figure 247, for all groups of designs, there is surprisingly little change in the number of designs with rim-type integration cores. There are absolutely no designs with rim-type integration cores among the groups of designs generated by the operations of linking vertex to vertex, linking vertex to edge, and linking edge to edge. On the other hand, the operation of disjoining vertices yields the highest number of designs with rim-type integration cores, followed by the mixed operations and the operation of splitting vertex.

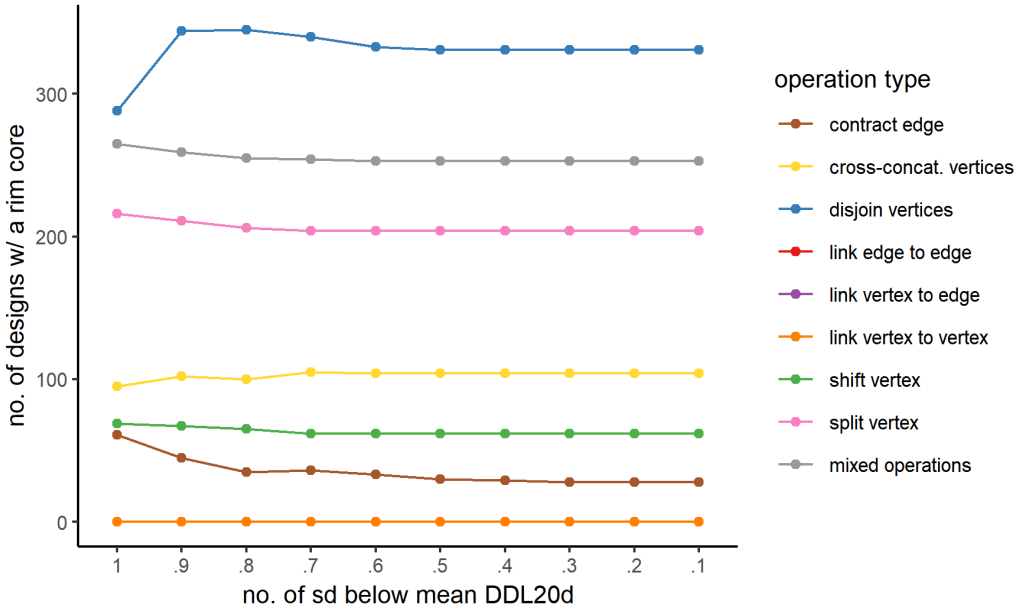

Figure 247. A line graph showing the change in the number of designs that have a rim-type integration core as the criterion of what makes up the integration core is gradually relaxed.

Examples of designs with a rim-type integration core are shown in Figure 248. In generating the examples, we define the integration core as the set of continuity lines whose DDL20d values are at least 0.5 standard deviations below the mean DDL20d.

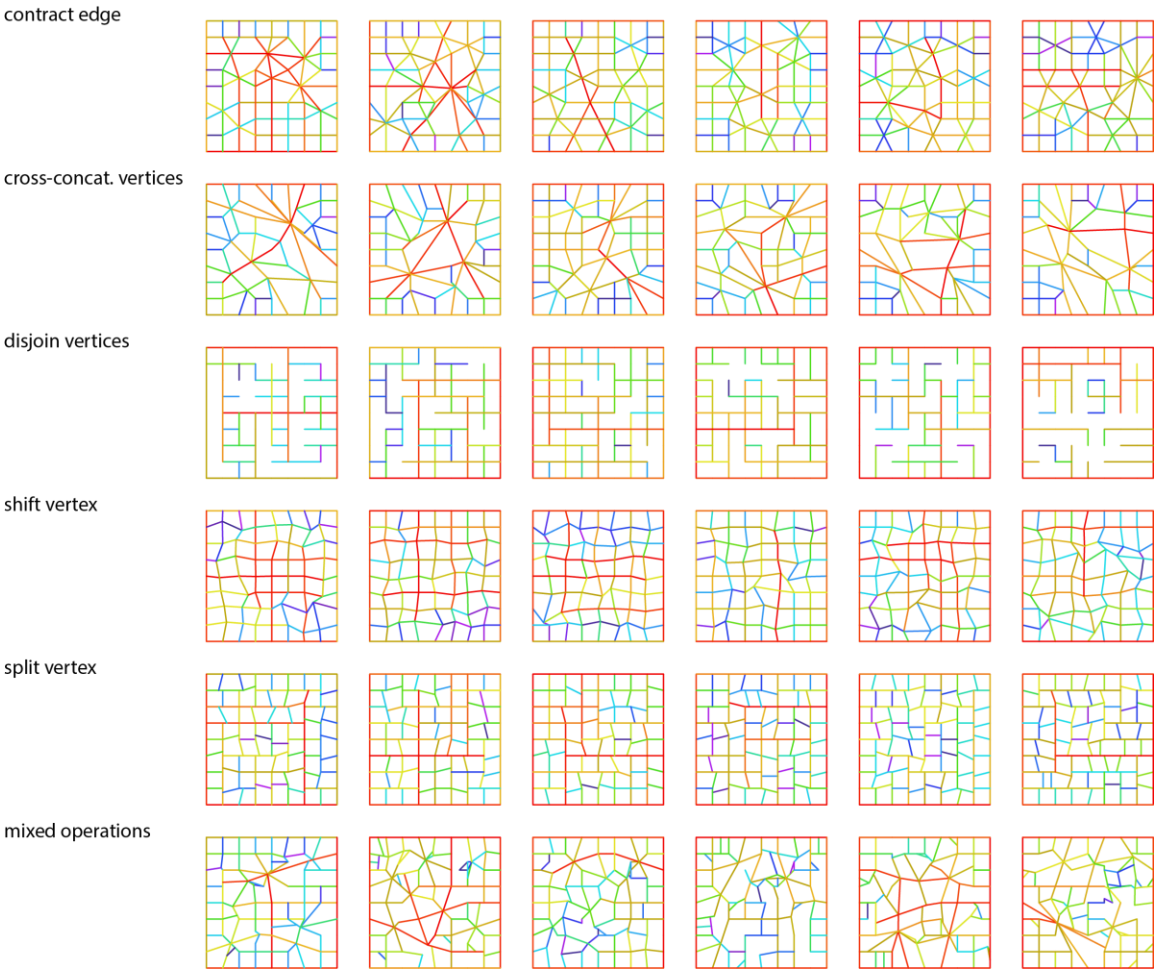

Figure 248. Examples of designs with rim-type integration cores selected from each group of designs.

11.2.4 Clarity of type

Among a particular group of designs that share a common type of integration core, further queries can be conducted based on additional criteria. This section shows how the designs with full-deformed-wheel-type integration cores can be further differentiated.

The quintessential full-deformed-wheel-type integration core in our analysis takes six traversing continuity lines, with two of them crossing each other at the center of the superblock and the other four forming the boundary of the superblock. Therefore, the minimum size for a full-deformed-wheel-type integration core is $6 \times 800 = 4800$ m. The additional continuity lines included in the integration core often "blur" the pattern of the deformed wheel or make the wheel-structure less clear (for example, consider the designs generated by the three linking operations, the main grid altogether stands out as the integration core, making the deformed-wheel-form much less clear).

Therefore, we may distinguish between the designs whose integration core is small and clearly a deformed wheel and those whose integration core is large and thus we have to filter out some noise to see the deformed-wheel structure. We may define the *clarity* of the type of integration core observed in a specific design by the ratio between the minimal size of that type of integration core and the observed size of the integration core of the design. For example, based on the design space we created, we can define the clarity of the full-deformed-wheel-type integration core as the ratio between the minimal size of the deformed-wheel-type integration core (4800 m) and the observed size of the integration core in the design. Figure 249 shows examples of designs with the clearest full-deformed-wheel-type integration cores and the ones with most unclear full-deformed-wheel-type integration cores. In generating the examples, we define the integration core as the set of continuity lines whose DDL20d values are at least 0.5 standard deviations below the mean DDL20d.

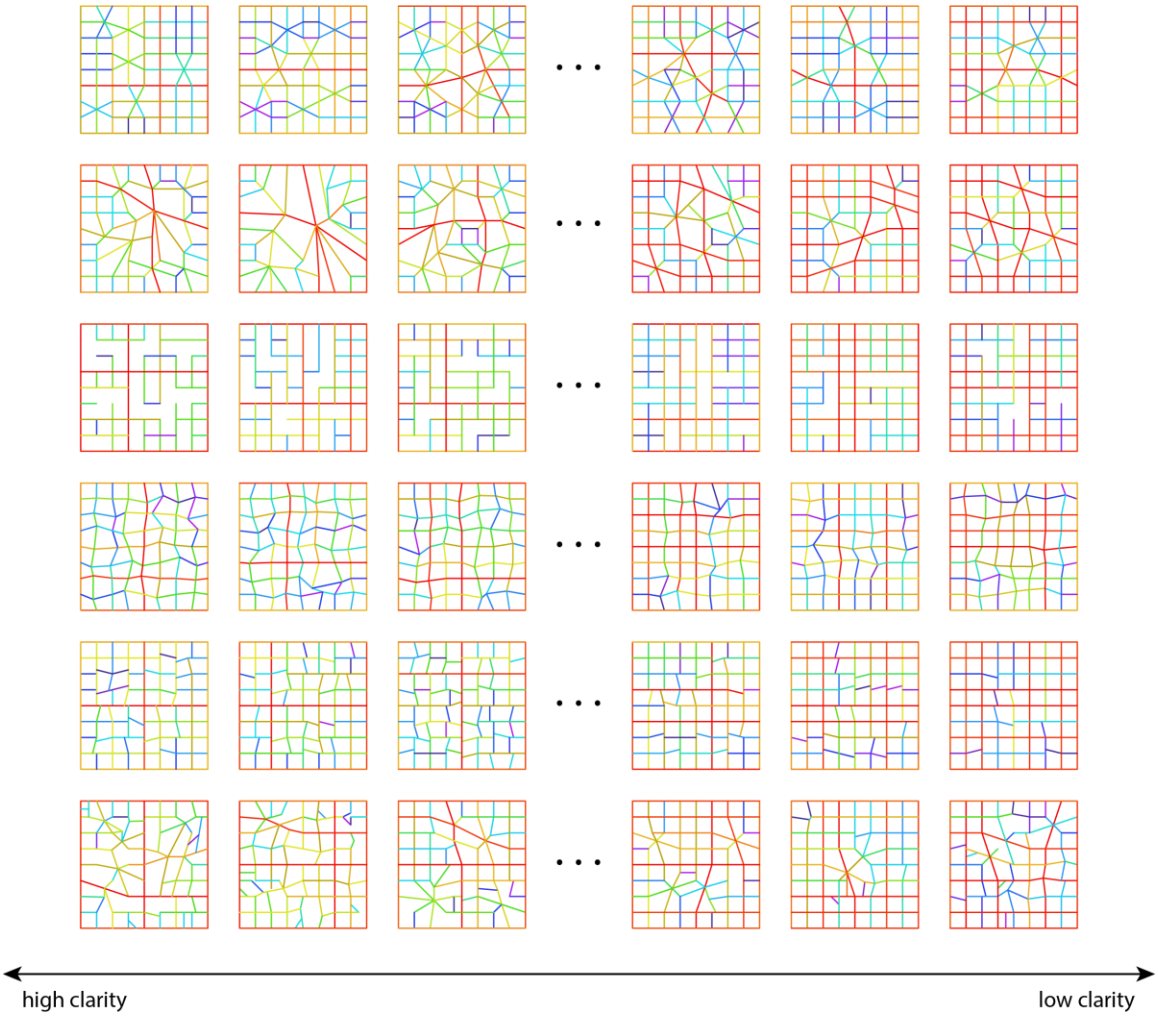

Figure 249. Designs with full-deformed-wheel-type integration cores. Designs in different rows are generated by different operations.

11.3 Discussion

11.3.1 The different potentials for developing different types of integration cores

By parametrically defining integration cores based on how much the mean DDL value of the core differs from the mean DDL value for the entire street network, we are able to study the interplay between the size and the shape of the integration core for each street network. The findings can be summarized as follows.

1. As the criterion of what makes up the integration core is gradually relaxed, the rate of growth of the integration core size—whether relativized against the total street length or not—reflects the distribution of DDL values in a design. Designs in which the primary street network is highly differentiated from the rest of the system yet internally undifferentiated in terms of centrality (measured by DDL) exhibited step-like growth trajectories. Designs in which streets are sufficiently differentiated across the full spectrum of centrality exhibited ramp-like growth trajectories.
2. Surprisingly, we could generate designs that have the full-deformed-wheel-type integration core (or more generally, the radial-type integration core) by applying *any* type of the operations being studied. While in most cases, the type of operation we choose, in and of itself, does not determine the specific type of integration core to be formed in a design, it does seem to affect our chances in getting designs with specific types of integration cores—provided that consistent criteria are adopted regarding what makes up the integration core. For example, it is more likely to yield a full-deformed-wheel type integration core by using the three linking operations when the criterion of what makes up the integration core is not too strict. It is also more likely to yield a rim-type integration core by disjoining vertices.

To understand the different potentials for developing specific types of integration cores, it helps to reflect on the following two questions:

- Is the differentiation caused by each operation widely spread or narrowly focused?

- Does the operation tend to conserve or produce long traversing streets in the street network?

The first question is about the scope of the impact caused by a specific operation. A widely spread differentiation in centrality (measured by DDL) implies that significant changes of DDL values can be observed across a very large portion of the street network, while a narrowly focused differentiation in centrality implies that significant changes of DDL values can only be observed within a very limited area of the street network, with the vast portion of the street network largely unaffected (either in terms of absolute values or changes in ranking). For example, the differentiation in centrality caused by the three linking operations (especially linking edge to edge) is very narrowly focused: the local infill streets have very little impact on the global pattern of centrality determined by the primary street network (i.e., the horizontal and vertical streets inherited from the initial square grid design). In many cases, the narrowly focused differentiation caused by local infills foregrounds the confrontation between the global and local scales. Regarding this point, a detailed explanation of why the global pattern of centrality established by a regular grid is hard to destruct by local insert streets can be found in our study of the supergrid (Peponis et al., 2015).

The second question is about the probability of forming long traversing streets (represented by continuity lines in the street graph) inside the boundary of the superblock design. The presence, or absence, of internal long traversing continuity lines has much to do with the identification of specific types of integration cores. For example, disjoining vertices and splitting vertex clearly tend to break existing long traversing continuity lines inside the superblock, which, in turn, increase the probability of forming rim-type

integration cores (Figure 247). On the contrary, the three linking operations never break existing long traversing continuity lines. As a result, starting from a square-grid design, they never produce designs with rim-type integration cores.

11.3.2 Description retrieval

It is important to notice that two fundamentally different sets of rules have been presented here, one associated with the *generation* of the street networks, while the other associated with the *description* of the street networks. We can sort the universe of designs into different types either based on the generative principles they embody (i.e., the different types of operations or generative rules) or the syntactic principles we generalized (i.e., the different syntactic types defined based on the patterns of centrality). The latter is essentially related to what Hillier and Hanson (1984) called the act of "description retrieval". According to them, locally ruled processes (such as ordered collective activities in human society) can give rise to new levels of order, and the emergent global order is knowable and retrievable by reflection. They argued that the description-retrieval mechanism is important for reproducing the same global order but without going through the same local processes. This underscores the importance to distinguish between the generative principles and the syntactic principles, or what we may simply call the "production rule" and the "reproduction rule". As we have found, there is no guarantee that by adopting a particular type of generative rule (i.e., production rule) we are bound to arrive at a particular syntactic type (which may further be used as the reproduction rule). In other words, setting off a generative process driven by local rules—however defined—does not necessarily lead to a global order that is desired. This has two implications. First, we cannot simply assume that we have always accurately

reproduced what we have built by just applying the same locally defined rules. Constant reflection on the emergent global order is important. Second, unless we can precisely and intelligently describe the global order (such as the global pattern of centrality studied in this chapter) arising from the numerous individual activities carried out at the local level, we are not sure that we have precisely reproduced the desired global order that was built into the things that we had built.

11.3.3 A word about the realization of the global order—Or, from syntactic types back to generative principles

Finally, we make a few remarks on the implementation of the global order. Suppose that we intend to design a street network that has a full-deformed-wheel-type integration core, how are we going to materialize that design intention?

It turns out that there are many ways to meet such a design intention. Figure 250 shows schematic superblock designs that can give rise to a deformed-wheel-type integration core, inspired by the universe of designs generated by our algorithms. However, it should be noted that, although the same global order is realized in each case, the different arrangements of local streets offer, or deny, certain opportunities for urban development, and can lead to drastically different experiences of the urban environment.

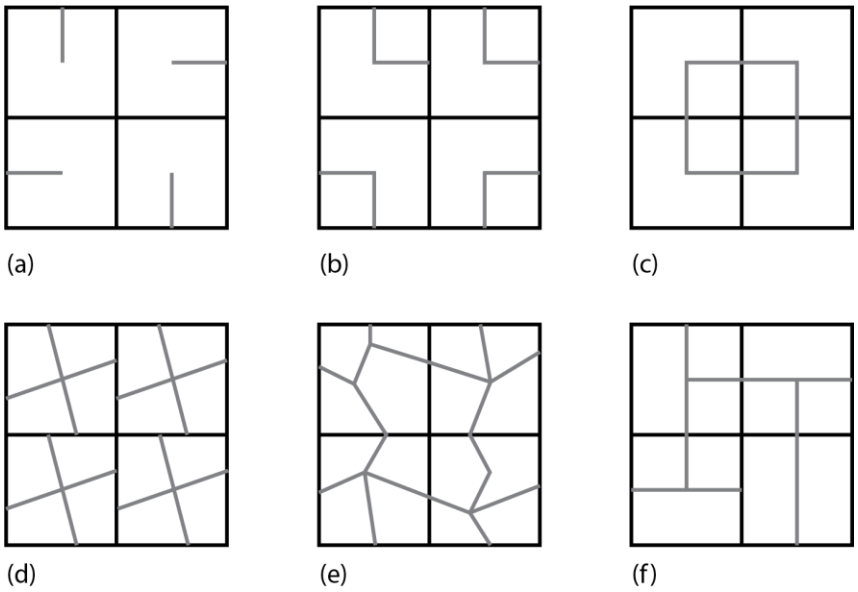

Figure 250. Schematic superblock designs that can give rise to a full-deformed-wheel-type integration core.

The schematic designs were all created by first imposing a global structure that resembles a full deformed wheel with two internal main traversing streets crossing each other at the center and four bounding streets at the edges (Figure 250). The local streets were added later. The first design, with the cul-de-sacs hanging on the main street network, suggest a clear distinction between “mobility” and “access”. The cul-de-sacs, by definition, do not bridge existing streets, therefore contributing to a closed, introverted environment (Figure 250a). The second design substitutes loops for cul-de-sacs. However, the loops are placed independently of each other and there are no intersections formed by the local streets. This design foregrounds the confrontation between the global and the local scale without projecting any intensity of experience at the local scale (Figure 250b). The third design aligns the individual loops with each other and thus forms a continuous internal ring street that exists independently of the main street network. By walking along the internal ring street, one periodically confronts the main

streets (Figure 250c). The fourth design has further developed local street networks inside the original blocks divided by the two internal main streets. The intersections inside the subblocks can potentially become points of intense development and provide heightened experience at the local scale. However, these local street networks are deliberately misaligned, suggesting a particular mode of cellular development (Figure 250d). In the fifth design, the global pattern of centrality is shaped by varying the geometric alignment of streets: some are straight, some are sinuous. However, the overall connectivity is high. The experience gained by moving about such systems has more to do with the differentiated street views and speeds of movement moderated by the curvature of the streets (Figure 250e). The last design is an interesting one (Figure 250f). The street network is rich differentiated. There are a variety of street lengths: some are long, some are short, some are in the middle range. Moreover, streets are connected with one another in different ways: there are long streets connecting long streets with moderately long streets, long streets connecting moderately long streets with short streets, moderately long streets connecting moderately long streets with short streets, short streets connecting long streets with long streets, and so on. Accompanying the great variety of length and connection is the richly differentiated syntactic conditions, which, in turn, provide the soil for diverse types and scales of development that would contribute to a vibrant and intense urban experience. The last design serves as an example of the kind of street network that is not only densely connected but also continuous and continuously differentiated.

How to exactly model the generative rules that give rise to the specified global pattern of centrality as well as the specific qualities in the arrangement of local streets

discussed above is beyond the scope of this dissertation. However, the findings on the relationship between the generative principles and the syntactic outcomes (in terms of syntactic measures and syntactic types) discussed this chapter and the previous chapter provide some insights in that direction.

CHAPTER 12
CONTRIBUTION, EXTENSIONS, AND LIMITATIONS

12.1 Contribution

This thesis focused on understanding differentiation defined in two ways. First, as variability of local conditions; second, as an emergent global structure of centrality. The methodology and computational tools developed also illustrate how the interplay between generative programs, analytical programs and query programs can provide a framework for systematically exploring and formulating more discriminating ideas about spatial structure, ideas that can also inform design intention and design evaluation.

12.1.1 Our work in relation to Hillier's work

(1) We studied dynamical spatial processes in which sequences of local spatial interventions give rise to structures of differentiation defined at the global scale. In the field of space syntax, these have not been pursued actively after Hillier started this line of research (Hillier, 1996b). In "Chapter eight: Is architecture an *ars combinatoria*?" of his book, he conducted a series of theoretical experiments to study the global effect caused by individual, local spatial moves—specifically, the incurred overall gain or loss of depth (a generic measure indicating the distance between two spaces, however the distance itself is defined) for the whole system. Architecturally, the "initial premise" that Hillier started with can be constructed by placing square rooms (which he called "cells") of uniform size next to each other so that each room is surrounded by four rooms, with a door placed in the center of each partition

wall that separates one room from another. The doors are all open initially. The distance between any pair of rooms next to each other is uniformly one unit. This configuration can thus be represented by a *square grid graph* where a vertex represents a room and an edge indicates that two rooms are adjacent and connected to each other. The main local moves that Hillier played with, when interpreted architecturally, are about closing and opening the doors between the rooms, or, when interpreted in the context of the underlying graph, about removing and adding edges between vertices. Of course, the local moves defined based on the underlying graph cannot violate the architectural reality. For example, a vertex cannot be linked to more than four vertices—because based on the initial "building layout" just described, a square room can at most be open to four square rooms, with each placed on one side. Likewise, edges that can potentially create shortcuts between vertices are not really possible because in a grid layout, one can only move in the four cardinal directions. While the initial premise we used for deformation is also a square grid, we do not have the constraints built into the architectural representation used by Hillier. The square grid graphs we use are conceived as *street graphs*. Street graphs represent street patterns, and as such they impose much fewer restrictions upon the local moves that can be used to deform the street graph—therefore, we have a much wider set of operations and rules compared to those deployed by Hillier. In the street graphs we studied, a vertex can have more than four neighbors as we can find street intersections where five or more streets converge. Edges are weighted by their length;

therefore, shortcuts are explicitly acknowledged. The different geometric alignment of edges are not only allowed but also treated as a fundamental property to be analyzed: people, of course, do not only move along cardinally oriented axes in cities—they move in all directions depending on street alignment, and the difference in the degree of path rotation along the course of their movement matters.

(2) The focus of Hillier's work, as presented in "Part three: The laws of the field" of his book "Space is the Machine", was on understanding why buildings and cities have certain formal characteristics that are "nearly-invariant" despite the influences exercised by different societies and cultures. Thus, the purpose of his theoretical experiments was to show how the commonalities exhibited in the emergent configurational properties of the wide range of cases studied are explained by the need to fulfill generic functions related to circulation and intelligibility via a dynamic spatial process governed by "local-to-global spatial laws". In other words, his real interest lay in the *convergence* of the emergent properties. Perhaps for that reason, the local moves he used are "strategic" moves to achieve particular ends: *maximize* or *minimize* the overall depth gain for the whole system under study. Furthermore, in the experimental generative process, and subsequent moves take into account the condition created by prior moves. In that sense, his real focus is not on the local moves per se, but rather on the strategy involved in deploying those local moves—hence the thing he studied is best described as *local maneuver* instead of *local move*. By contrast, we are as interested in *divergence* as in convergence. The

rules developed based on the individual syntactic operators were simply meant to enable repeated applications of each operation with least restrictions as possible. They are not intended to *optimize* anything. Our aim is not to understand the convergent properties observed in empirical samples of interest. Instead, we try to understand the theoretical design space within which interesting architectural possibilities can exist by studying the structures and types of differentiation in a large sample of theoretical designs.

12.1.2 Main contribution and key findings

(1) In the field of space syntax, the differentiation among street networks is almost always discussed from the perspective of depth gains (which, in turn, is often indicated by the measure of integration or closeness centrality). We, however, propose that differentiation is a *multi-dimensional* idea. We show that the different dimensions of differentiation are not only interrelated, but their relationship can be quantified. Moreover, while the relationship between different dimensions of differentiation usually has a consistent direction, its slope can vary, depending on the type of operation used to create the differentiation. The identification of such underlying trends further reinforces the importance of discriminating between different patterns of differentiation and different ways of bringing differentiation about by deforming a regular grid. In practical terms, differentiation along a certain dimension may be interpreted as a benefit or a cost. For example, while increasing the diversity of syntactic conditions inside a street network may be considered desirable, the creation of highly differentiated yet oddly-shaped blocks or highly

differentiated yet segregated areas is often unwanted. The variation in slope thus suggests that properties that may be desirable can be achieved with varying costs regarding the properties that may be undesirable. The trends analysis we conducted shows that, in principle, a balanced (or even optimized) design can be achieved by making consistent, carefully chosen, small design moves as would be suggested by Hillier's approach, or by selecting appropriate operations.

(2) In the field of space syntax, the distribution of integration is considered as important as the degree of integration. As much space syntax literature has shown, a differentiated distribution of integration suggests the presence of structural bias, which, in turn, suggests different perceptive, cognitive, and functional affordances. The distribution of integration is usually studied based on the shape and size of the integration core. However, the definitions of certain types of integration cores frequently mentioned in the literature remain largely intuitive. As part of the effort to query the design space, we developed rigorous, algorithmic definitions for different types of integration cores. For example, the deformed-wheel-type integration core has been defined intuitively or heuristically in the space syntax literature, but now we have given a precise definition. By developing and refining the algorithmic definitions, query algorithms can be iteratively applied to confirm syntactic types already known but also discover interesting (and possibly generalizable new) types that have not yet received attention. The methodology and computational framework we developed thus provides a way to engage both

intuition and rigor, and prompts us to define, refine and re-define types of designs.

(3) The syntactic operators used to generate designs do *not* necessarily lead to specific emergent global properties of the street network of the superblock. Although we cannot accurately *predict* the specific type we get by applying a specific generative rule (that involves a specific syntactic operator), we did find that by applying certain generative rules, we are *more likely* to generate designs of a specific syntactic type—provided that consistent criteria are adopted regarding what makes up the integration core. For example, a deformed-wheel-type integration core is more likely if the three linking operations are used, provided that the criterion of what makes up the integration core is not too strict. However, the linking operations are not effective in creating spine-type integration cores. By contrast, designs with spine-type integration cores are more likely if we contract edges or cross-concatenate vertices. Disjoining vertices in a square grid design tends to reduce the number of traversing streets, thus making rim-type integration cores more likely. A promising path for future research is to explore ways of combining rules or operations that are more likely to generate designs of certain desirable types of integration cores—for example, designs with partial-radial-type, or better, deformed-wheel-type integration cores.

12.2 Types of Integration Cores Observed in Well-Known Conceptual Schemes and Real-Life Examples

The analysis of the sizes and types of integration core developed in the previous chapter can readily be applied to some of the well-known conceptual schemes and real examples across the world. We illustrate this point by studying a total of ten superblock designs, including the design of a New York neighborhood proposed by Perry and Whitten in 1929, the design of Sector 19 of Chandigarh—finalized by Le Corbusier and his team in the 1950s, the design of Sector G-7 of Islamabad— proposed by Doxiadis and completed in the 1960s, two superblocks located in the central district of Beijing, two superblocks located in Gangnam District, Seoul, and three superblocks from the US—located in Chicago, Los Angeles, and Phoenix, respectively.

As shown in Figure 251, the rectangular superblocks vary moderately in size, with the side length ranging from slightly below half a mile to just over one mile. The internal street networks are different in terms of street density and geometric properties. Some are quite dense, such as those in Gangnam, while some are a lot sparser, such as the one in Phoenix. Some are grid-like with streets running perpendicular to each other, such as the one in Chicago, while some are dominated by curvilinear streets with various angles of incidence, such as Perry and Whitten's proposal.

As shown in Figure 252, there also seem to be a variety of patterns of centrality. Here, we focus on comparing the sizes and types of the integration core. Figure 253 shows the growth trajectory of the size of the integration core as the criterion of what makes up an integration core is gradually relaxed for each design. Clearly, the absolute

size of the integration core is positively influenced by the overall size of the design. This becomes more and more manifest as we relax more and more the criterion of what makes up the integration core. For example, as shown in Figure 253A, the designs which have considerably larger integration cores in terms of absolute size are also the ones which have the greatest total street length—most notably Sector G-7 of Islamabad, followed by the three superblocks located in the US. Relativizing the size of the integration core against the total street length of each design yields a quite different picture. For example, as shown in Figure 253B, while Sector G-7 of Islamabad has the largest integration core in terms of absolute size, its relativized core size is only in the mid-range. By contrast, the two superblocks from Beijing whose absolute core sizes were ranked among the lowest now score better than most other designs when we relax the criterion by 0.6 standard deviations or more.

The growth trajectories of the size of the integration core also give us a clue about the distribution of DDL20d values in each design. In the cases of superblocks in Beijing and Chicago, the growth trajectories are more "step-like", suggesting that a large proportion of the street network are highly connected but also somewhat undifferentiated as commonly observed in designs in which the primary road network resembles a regular grid.

Figure 251. A sample of superblock designs.

Figure 252. Patterns of centrality as measured by DDL20d. The spectrum from red to blue corresponds to the range from low to high DDL20d.

On the other hand, in cases such as the superblocks in Gangnam or in Phoenix, the growth trajectories are much more gentle and "ramp-like". Interestingly, the two superblocks in Beijing share lots of similarities in the growth trajectories, as do the two superblocks in Gangnam, which seem to suggest that the growth trajectories tend to be similar for superblocks that share the same historical or cultural origin. However, the reverse is certainly not true. For example, although the growth trajectory for the superblock in Phoenix shares striking similarities with those of Gangnam, they obviously differ not only in historical and cultural origins but also in connectivity and density.

The algorithm used to identify the types of integration cores in the previous chapter is also applied to study the above ten examples. Because of the limitation of the current implementation of the algorithm, we have slightly adjusted the boundary for a few superblock designs to make them a strict rectangle, including the two superblocks in Beijing, Sector 19 of Chandigarh, and Perry and Whitten's neighborhood design (Figure 252). The integration core is defined as the set of continuity lines whose DDL20d values are at least 0.5 standard deviations below the mean DDL20d value for each design. For the traversing continuity lines within the integration core, we call the ones inside the boundary "spines" and the ones on the boundary "rims". The intersections between the spines are called "foci", and the foci which lie near the geometric center of the design are called "hubs". In this analysis, only the intersections that lie within the radius of a quarter of the average side length of the superblock from its center are considered to be hubs.

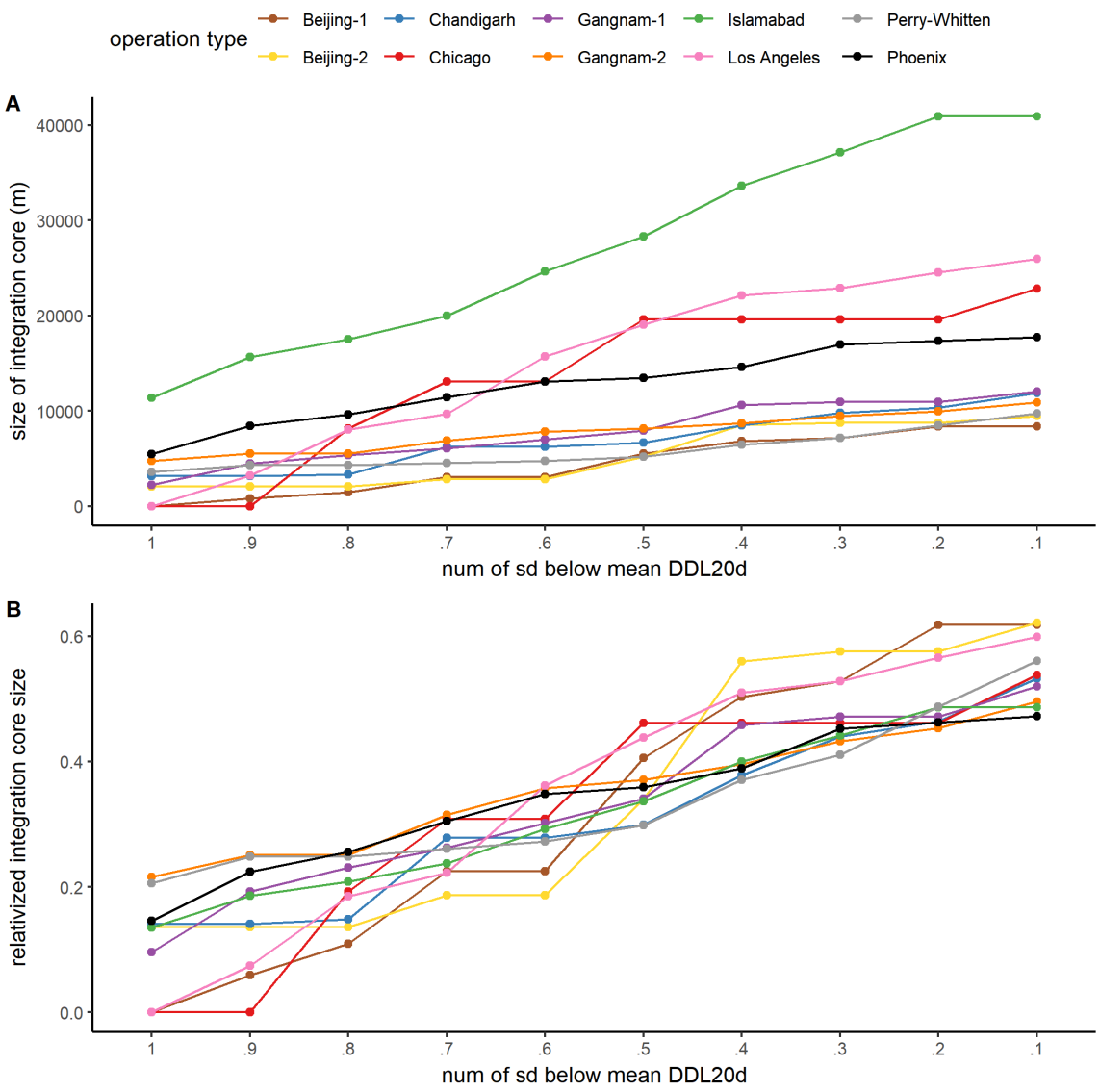

Figure 253. Growth trajectories of the size of the integration core as the criterion of what makes up an integration core is gradually relaxed. A: The absolute size of the integration core. B: The relativized size of the integration core.

Table 4. Number of hubs, foci, rims, and spines in each superblock design (α = 20°)

Superblock name	*No. of hubs*	*No. of foci*	*No. of rims*	*No. of spines*
Beijing-1	0	0	3	2
Beijing-2	0	0	3	3
Chandigarh	0	0	2	1
Chicago	1	14	3	9
Gangnam-1	0	0	3	1
Gangnam-2	0	0	4	1
Islamabad	0	0	4	0
Los Angeles	1	1	4	2
Perry-Whitten	0	0	4	0
Phoenix	0	0	3	0

Table 4 shows the counts of hubs, foci, spines, and rims for each superblock design, with the threshold angle (α) set to 20°. All types of integration cores described in the previous chapter are identified among the ten superblocks studied here. Let us denote the counts of hubs, foci, rims, and spines in a design by N (hubs), N (foci), N (rims), and N (spines), respectively.

(1) Radial-type integration core

The only condition for being a radial-type integration core is N (hubs) > 0. Among the ten superblocks, only the superblocks in Chicago and Los Angeles have radial-type integration cores. Moreover, although both have a single hub in the design, the superblock in Chicago has considerably more foci (a total of 14

foci) than the one in Los Angeles (only a single focus), suggesting quite different urban experience.

(2) Full-deformed-wheel-type integration core

For a rectangular superblock design, the conditions for being a full-deformed-wheel-type integration core are N (hubs) > 0 and N (rims) = 4. Therefore, only the superblock in Los Angeles satisfies both conditions and considered to have a full-deformed-wheel-type integration core.

(3) Spine-type integration core

The conditions for being a spine-type integration core are N (spines) > 0 and N (foci) = 0. Therefore, the two superblocks in Beijing, Sector 19 of Chandigarh, and the two superblocks in Gangnam are considered to have spine-type integration cores.

(4) Rim-type integration core

The conditions for being a rim-type integration core are N (spines) = 0 and N (rims) > 0. Rim-type integration cores are identified in the design of Sector G-7 of Islamabad, the neighborhood design proposed by Perry and Whitten, and the superblock in Phoenix.

The classification of the superblock designs generally speak to our intuition. For example, it seems appropriate to say that the superblocks in Beijing have spine-type integration cores, because the spines identified in our analysis clearly correspond to the east-west running streets or alleyways (called *hutong* in Chinese), which is associated with a key organizing principle of the urban fabric in Beijing. Likewise, it is not surprising to see Perry and Whitten's neighborhood design or Doxiadis's design for

Islamabad be considered to have a rim-type integration core, because they were intended to be built more or less as self-contained units or sectors primarily for residential buildings.

However, classifying the integration cores of the two superblocks in Gangnam as "spine-type" cores does not speak to our intuition. A spine-type core implies that the most integrated continuity lines all run in similar directions (such as the superblocks in Beijing) or a highly integrated main road cutting through the superblock with minor roads branching off from it (such as Sector 19 of Chandigarh). But neither is the case for the superblocks in Gangnam. Inside the Gangnam superblocks, the street network does not suggest a predominant direction, nor is it dendritic. On the contrary, it is more like a warped grid with diverse degrees of centrality created via connection, offset, or rotation of streets. The various angles of incidence formed by streets in the superblocks suggest that a change of the threshold angle could have a major impact on the analysis.

As shown in Table 5, when we increase the threshold angle (α) from 20° to 35°, more traversing continuity lines are recognized in the Gangnam superblocks. The Gangnam-1 superblock now has two hubs identified, the most among all cases. Now it is considered to have a radial-type integration core, or a partial-deformed-wheel-type core. Likewise, three more spines and two foci have been identified in the Gangnam-2 superblock. It should be noted that the counts of hubs, foci, spines, and rims remain the same for all the other cases. The increase of the threshold angle only made a difference in the case of the two Gangnam superblocks.

Table 5. Number of hubs, foci, rims, and spines in each superblock design (α = 35°)

Superblock name	*No. of hubs*	*No. of foci*	*No. of rims*	*No. of spines*
Beijing-1	0	0	3	2
Beijing-2	0	0	3	3
Chandigarh	0	0	2	1
Chicago	1	14	3	9
Gangnam-1	2*	3*	3	4*
Gangnam-2	0	2*	4	4*
Islamabad	0	0	4	0
Los Angeles	1	1	4	2
Perry-Whitten	0	0	4	0
Phoenix	0	0	3	0

Note. * Different from the number yielded from the previous analysis where α = 20°.

Thus, the algorithms developed for the analysis of the automatically generated theoretical designs can also be used to analyze the real-world examples, provided that the superblocks are rectangular, and a sensible threshold angle is used. In the field of space syntax, the patterns of the integration cores are often discussed informally and limited to verbal descriptions (e.g., “deformed wheel”). Thus, the formal, algorithmic definition of the different types of integration cores given here allows a more systematic and rigorous identification of syntactic types.

Perhaps even more important is that a rigorous definition of syntactic types not only enriches the description of designs but encourages further inspection and refection

of design intentions as well. This can be done in at least two ways: (a) identification of subtypes (for example, the "full-deformed-wheel-type integration core" can be viewed as a special case, or a sub-type of the "radial-type") and (b) reflection on the different mechanisms that generate the same type of designs. To illustrate the latter point, we can compare the three superblock designs which all have a radial-type integration core based on the above analysis (with $\alpha = 35°$). While they all have a radial-type integration core, the mechanisms involved (and the associated design intentions) are quite different. In the case of the superblock in Chicago, the radial-type integration core is a natural result of a uniform, repetitive orthogonal grid, seeming to promote an egalitarian distribution and use of land (at least at face value); in the case of the superblock in Los Angeles, the radial-type (or full-deformed-wheel-type) integration core is very much the product of the "anti-grid" design gestures observed in three of the four quadrants of the superblock (however, whether the "radials" are acting as seams joining the local areas or as dividers is questionable); in the case of the superblocks in Gangnam, the radial-type integration cores are results of highly connected yet richly differentiated street grids and act as dynamic interfaces between different scales of movement and development. After all, it is the understanding of the mechanisms that lead to same or different types of designs that equip us with the necessary design knowledge required to both discover and tackle new problems.

12.3 Limitations

There remain some limitations regarding the query rules/algorithms we used to define and identify particular syntactic types and the generative rules/algorithms we used to develop the universe of designs.

12.3.1 Query rules/algorithms

The current query algorithms used to identify the different types of integration cores require that the study area strictly resembles a square or a rectangle. Future work could address this limitation to enable identification of different types of integration cores even when the study area has more than four sides and curvy perimeters. Besides, more refined query algorithms can be developed in the future to derive important subtypes of designs.

Another limitation is associated with the proper choice of a threshold angle to determine a change of direction. As we have seen in the analysis of the Gangnam superblocks, varying the threshold angle could have a significant impact on the query results. While the impact is almost negligible for street networks where most of the streets intersect with one another at right angles (or approximately 90°), it can significantly affect the result for street networks in which curvilinear streets are prevalent and streets intersect with each other at various angles. The "tuning" of threshold angle to best characterize a given street network needs to be backed by further research.

12.3.2 Generative rules/algorithms

In this study, the effects of the generative rules are always tested based on the deformation of a uniform grid. Future work could further test the effects of the generative rules by applying them to street networks that are significantly different from a uniform grid. Besides, in our study, certain types of operations are parametrically defined, yet we only studied them with fairly restricted parameter settings. We also tried to make the generative rules/principles as simple as possible so that the operations can be applied

with as few restrictions as possible. In that sense, the descriptions of the generative rules themselves are relatively light. In the future, more complex generative rules (hence heavier descriptions) can be modeled to explore the potential for guiding the production of designs toward particular types of street networks.

Finally, it should be acknowledged that the designs generated by our algorithms are diagrammatic and not full-fledged urban design schemes. The segments in the street networks we generated are more like street centerlines. Creating realistic urban layouts from the street centerline diagrams requires additional work, such as assigning street widths to the centerlines, or adjusting the shape of the blocks to avoid sharp angles. (In the architectural field, research aimed at generating more realistic urban layouts can be seen in works by, for example, Duarte and Beirão (2011), and Miao et al. (2018).) Nevertheless, theoretical layouts in the design universe shown here prove to be quite useful in developing, questioning, and modifying intuitions that arise as we study real examples. Moreover, the street networks are the backbone upon which we design and build our urban life. The knowledge gained by rigorous study of how local arrangements of streets affect the global properties of a street network can be tremendously useful to urban designers and policymakers in deriving general principles for the organization of streets.

REFERENCES

American Association of State Highway and Transportation Officials [AASHTO]. (2001). *A policy on geometric design of highways and streets (4th ed.)*. Washington, D.C.: Author Retrieved from http://nacto.org/docs/usdg/geometric_design_highways_and_streets_aashto.pdf.

Aristotle. (trans. 1932). Aristotle: Politics (H. Rackham, Trans.). In. Cambridge, MA: Harvard University Press.

Asami, Y., Kubat, A. S., & Istek, C. (2001). Characterization of the street networks in the traditional Turkish urban form. *Environment and Planning B: Planning and Design, 28*(5), 777–795.

Bacon, E. N. (1976). *Design of cities (Revised edition)*. New York: Penguin Books.

Bared, J. G., & Kaisar, E. I. (2001). *Advantages of offset T-intersections with guidelines.* Paper presented at the 12th International Conference: Road Safety on Three Continents, Moscow, Russia.

Barthelemy, M. (2011). Spatial networks. *Physics Reports, 499*(1), 1-101. doi:https://doi.org/10.1016/j.physrep.2010.11.002

Batty, M. (2001). Exploring isovist fields: Space and shape in architectural and urban morphology. *Environment and Planning B: Planning and Design, 28*(1), 123-150. doi:10.1068/b2725

Boardman, J. (Ed.) (1994). *The Cambridge ancient history: Plates to Volume V and VI*. Cambridge, UK: Cambridge University Press.

Boeing, G. (2018). Measuring the complexity of urban form and design. *Urban Design International, 23*(4), 281-292. doi:10.1057/s41289-018-0072-1

Braid, I. C., Hillyard, R. C., & Stroud, I. A. (1980). Stepwise construction of polyhedra in geometric modeling. In K. W. Brodlie (Ed.), *Mathematical methods in computer graphics and design* (pp. 123–141). New York/London: Academic Press.

Chen, G., Esch, G., Wonka, P., Müller, P., & Zhang, E. (2008). Interactive procedural street modeling. *ACM Transactions on Graphics (TOG), 27*(3), 103.

Chiaradia, A., Hillier, B., Schwander, C., & Wedderburn, M. (2012). Compositional and urban form effects on centres in Greater London. *Proceedings of the Institution of Civil Engineers - Urban Design and Planning, 165*(1), 21-42. doi:10.1680/udap.2012.165.1.21

Christova, P., Scoppa, M., Peponis, J., & Georgopoulos, A. P. (2012). Exploring small city maps. *Experimental Brain Research, 223*(2), 207–217.

Colquhoun, A. (1981). The superblock. In *Essays in architectural criticism: Modern architecture and historical change* (pp. 83–103). Cambridge, MA: The MIT Press.

Courtat, T., Gloaguen, C., & Douady, S. (2011). Mathematics and morphogenesis of cities: A geometrical approach. *Physical Review E, 83*(3 Pt 2), 036106. doi:10.1103/PhysRevE.83.036106

Crucitti, P., Latora, V., & Porta, S. (2006). Centrality measures in spatial networks of urban streets. *Physical Review E, 73*(3), 036125. doi:10.1103/PhysRevE.73.036125

Dalton, R. C. (2003). The secret is to follow your nose: Route path selection and angularity. *Environment and Behavior, 35*(1), 107–131.

Dalton, R. C., & Kirsan, C. (2008). Small-graph matching and building genotypes. *Environment and Planning B: Planning and Design, 35*(5), 810-830. doi:10.1068/b31136

Davidson, M., & Dolnick, F. (2004). *A planners dictionary (PAS 521/522)*: American Planning Association Planning Advisory Service.

de Berg, M., Cheong, O., van Kreveld, M., & Overmars, M. (2008). *Computational Geometry: Algorithms and applications* (3rd ed.). Berlin/Heidelberg, Germany: Springer-Verlag.

Duarte, J. P., & Beirão, J. (2011). Towards a Methodology for Flexible Urban Design: Designing with Urban Patterns and Shape Grammars. *Environment and Planning B: Planning and Design, 38*(5), 879-902. doi:10.1068/b37026

Eastman, C., & Waiter, K. (1979). *Geometric modeling using the Euler operators* (Research Report No. 78). Retrieved from Pittsburgh, PA:

Feng, C., & Peponis, J. (2018). The definition of syntactic types: The generation, analysis, and sorting of universes of superblock designs. *Environment and Planning B: Urban Analytics and City Science*. doi:10.1177/2399808318813576

Feng, C., & Zhang, W. (2017). Grasshopper Reach Analysis Toolkit: Interactive parametric syntactic analysis. In T. Heitor, M. Serra, J. P. Silva, M. Bacharel, & L. C. da Silva (Eds.), *Proceedings of the 11th International Space Syntax Symposium* (pp. 159.001–159.012). Lisbon, Portugal: Instituto Superior Técnico, Departamento de Engenharia Civil, Arquitetura e Georrecursos, Portugal.

Feng, C., & Zhang, W. (in press). Algorithms for the parametric analysis of metric, directional, and intersection reach. *Environment and Planning B: Urban Analytics and City Science*. doi:10.1177/2399808319827299

Figueiredo, L., & Amorim, L. (2005). Continuity lines in the axial system. In A. van Nes (Ed.), *Proceedings of the 5th Space Syntax Symposium* (Vol. 1, pp. 161–174). Amsterdam, The Netherlands: Techne Press.

Figueiredo, L., & Amorim, L. (2007). Decoding the urban grid: Or why cities are neither trees nor perfect grids. In A. S. Kubat, Ö. Ertekin, Y. İ. Güney, & E. Eyüboğlu (Eds.), *Proceedings of the 6th International Space Syntax Symposium* (pp. 006:001–006:016). Istanbul, Turkey: İTÜ, Faculty of Architecture.

Groth, P. (1981). Streetgrids as frameworks for urban variety. *Harvard Architectural Review, 2*(2), 68–75.

Handy, S., Paterson, R. G., & Butler, K. (2003). *Planning for street connectivity: Getting from here to there (Planning Advisory Service Report Number 515)*. Chicago: American Planning Association.

Haq, S. (2003). Investigating the syntax line: Configurational properties and cognitive correlates. *Environment and Planning B: Planning & Design, 30*(6), 841–863.

Haq, S., & Zimring, C. (2003). Just down the road a piece: The development of topological knowledge of building layouts. *Environment and Behavior, 35*(1), 132–160.

Hess, P. M. (1997). Measures of connectivity. *Places, 11*(2), 58–65.

Hess, P. M., Moudon, A. V., Snyder, M. C., & Stanilov, K. (1999). Site design and pedestrian travel. *Transportation Research Record, 1674*, 9–19.

Hillier, B. (1996a). Cities as movement economies. *Urban Design International, 1*(1), 41–60.

Hillier, B. (1996b). *Space is the machine: A configurational theory of architecture*. Cambridge, UK: Cambridge University Press.

Hillier, B. (1997). Cities as movement economies. In P. Droege (Ed.), *Intelligent environments* (pp. 295-344). Amsterdam: North-Holland.

Hillier, B. (1999). Centrality as a process: Accounting for attraction inequalities in deformed grids. *Urban Design International, 4*(3–4), 107–127.

Hillier, B. (2002). A theory of the city as object: Or, how spatial laws mediate the social construction of urban space. *Urban Design International, 7*(3–4), 153–179.

Hillier, B., Burdett, R., Peponis, J., & Penn, A. (1987). Creating life: Or, does Architecture determine anything? *Architecture et Comportement / Architecture and Behaviour, 3*(3), 233–250.

Hillier, B., & Hanson, J. (1984). *The social logic of space*. Cambridge, UK: Cambridge University Press.

Hillier, B., Hanson, J., Peponis, J., Hudson, J., & Burdett, R. (1983). Space syntax: A different urban perspective. *Architects' Journal, 178*(48), 47–63.

Hillier, B., Penn, A., Hanson, J., Grajewski, T., & Xu, J. (1993). Natural movement: Or, configuration and attraction in urban pedestrian movement. *Environment and Planning B: Planning and Design, 20*(1), 29–66.

Hochmair, H., & Frank, A. U. (2000). Influence of estimation errors on wayfinding-decisions in unknown street networks – analyzing the least-angle strategy. *Spatial Cognition and Computation, 2*(4), 283-313. doi:10.1023/a:1015566423907

Jacobs, J. (1961). *The death and life of great American cities*. New York: Random House, Inc.

Javadi, A.-H., Emo, B., Howard, L. R., Zisch, F. E., Yu, Y., Knight, R., . . . Spiers, H. J. (2017). Hippocampal and prefrontal processing of network topology to simulate the future. *Nature Communications, 8*, 14652. doi:10.1038/ncomms14652

Jiang, B., & Claramunt, C. (2004). Topological analysis of urban street networks. *Environment and Planning B: Planning and Design, 31*(1), 151-162. doi:10.1068/b306

Kan, H. Y., Forsyth, A., & Rowe, P. (2017). Redesigning China's superblock neighborhoods: Policies, opportunities and challenges. *Journal of Urban Design, 22*(6), 757–777.

Karimi, K. (1997). The spatial logic of organic cities in Iran and in the United Kingdom. In M. Major, L. Amorim, & F. Dufaux (Eds.), *Proceedings of the 1st International Space Syntax Symposium* (Vol. 1, pp. 05.01–05.17). London, UK: University College London.

Kim, Y. O., & Penn, A. (2004). Linking the spatial syntax of cognitive maps to the spatial syntax of the environment. *Environment and Behavior, 36*(4), 483–504.

Kirkley, A., Barbosa, H., Barthelemy, M., & Ghoshal, G. (2018). From the betweenness centrality in street networks to structural invariants in random planar graphs. *Nature Communications, 9*(1), 2501. doi:10.1038/s41467-018-04978-z

Kostof, S. (1991). *The city shaped: Urban patterns and meanings through history*. London, UK: Thames & Hudson Ltd.

Kuipers, B., Tecuci, D. G., & Stankiewicz, B. J. (2003). The skeleton in the cognitive map: A computational and empirical exploration. *Environment and Behavior, 35*(1), 81–106.

Louf, R., & Barthelemy, M. (2014). A typology of street patterns. *Journal of The Royal Society Interface, 11*(101), 20140924. doi:10.1098/rsif.2014.0924

Lu, D. (2006). *Remaking Chinese urban form: Modernity, scarcity and space, 1949–2005*. New York, USA: Routledge.

Mantyla, M. (1984). A note on the modeling space of Euler operators. *Computer Vision, Graphics, and Image Processing, 26*(1), 45–60.

Marshall, S. (2005). *Streets and patterns*. Abingdon, England: Spon Press.

Marshall, S. (2016). Line structure representation for road network analysis. *Journal of Transport and Land Use, 9*(1).

Marshall, S., Gil, J., Kropf, K., Tomko, M., & Figueiredo, L. (2018). Street network studies: From networks to models and their representations. *Networks and Spatial Economics*. doi:10.1007/s11067-018-9427-9

Marshall, W., & Garrick, N. (2010). Effect of street network design on walking and biking. *Transportation Research Record, 2198*(1), 103-115. doi:10.3141/2198-12

Mehaffy, M. W., Porta, S., & Romice, O. (2015). The "neighborhood unit" on trial: A case study in the impacts of urban morphology. *Journal of Urbanism: International Research on Placemaking and Urban Sustainability, 8*(2), 199–217.

Miao, Y., Koenig, R., Knecht, K., Konieva, K., Buš, P., & Chang, M.-C. (2018). Computational urban design prototyping: Interactive planning synthesis methods—a case study in Cape Town. *International Journal of Architectural Computing, 16*(3), 212-226. doi:10.1177/1478077118798395

Moneo, R. (1978). On typology. *Oppositions*(13), 22–45.

Morris, A. E. J. (1994). *History of urban form before the Industrial Revolutions*. Essex, England: Pearson Education Limited.

Ortiz-Chao, C., & Hillier, B. (2007). In search of patterns of land-use in Mexico City using logistic regression at the plot level. In A. S. Kubat, Ö. Ertekin, Y. I. Guney, & E. Eyüboğlu (Eds.), *Proceedings of the 6th International Space Syntax Symposium*. Istanbul, Turkey: ITU Faculty of Architectur.

Ozbil, A., Peponis, J., & Stone, B. (2011). Understanding the link between street connectivity, land use and pedestrian flows. *Urban Design International, 16*(2), 125–141.

Patricios, N. N. (2002). Urban design principles of the original neighborhood concepts. *Urban Morphology, 6*(1), 21–32.

Peng, C., Yang, Y., Bao, F., Fink, D., Yan, D., Wonka, P., & Mitra, N. J. (2016). Computational network design from functional specifications. *ACM Transactions on Graphics (TOG), 35*(4), 131.

Penn, A., Hillier, B., Banister, D., & Xu, J. (1998). Configurational modelling of urban movement networks. *Environment and Planning B: Planning and Design, 25*(1), 59–84.

Peponis, J. (2006). The city as search engine. *Elelef(*Special Issue: For the Right to the City), 37–50.

Peponis, J., Bafna, S., & Zhang, Z. (2008). The connectivity of streets: Reach and directional distance. *Environment and Planning B: Planning and Design, 35*(5), 881–901.

Peponis, J., & Feng, C. (2016). 开放城市与有关美好城市生活的空间句法[The open city and the space syntax of good urban life]. *New Architecture, 164*(1), 42–51.

Peponis, J., Feng, C., Green, D., Haynie, D., Kim, S. H., Sheng, Q., . . . Wang, H. (2015). Syntax and parametric analysis of superblock patterns. *The Journal of Space Syntax, 6*(1), 109–141.

Peponis, J., Feng, C., & Park, J. (2017). Diversity and scale in superblock design. *Urban Design — School of Architecture, Tsinghua University, 13*(5), 30–41.

Peponis, J., Hadjinikolaou, E., Livieratos, C., & Fatouros, D. A. (1989). The spatial core of urban culture. *Ekistics-the Problems and Science of Human Settlements, 56*(334/335), 43–55.

Peponis, J., Park, J., & Feng, C. (2016). The city as interface of scales: Gangnam urbanism. In S. H. Kim, E. Cinn, K. Ahn, S. Kim, I. Chung, D. E. Jeong, & R. Enos (Eds.), *The FAR game*. Seoul, Korea: SPACE Books.

Peponis, J., Ross, C., & Rashid, M. (1997). The structure of urban space, movement and co-presence: The case of Atlanta. *Geoforum, 28*(3–4), 341–358.

Peponis, J., Wineman, J., Bafna, S., Rashid, M., & Kim, S. H. (1998). On the generation of linear representations of spatial configuration. *Environment and Planning B: Planning and Design, 25*(4), 559–576.

Peponis, J., Zimring, C., & Choi, Y. K. (1990). Finding the building in wayfinding. *Environment and Behavior, 22*(5), 555–590.

Perry, C. A. (1929). The neighborhood unit: A scheme of arrangement for the family-life community. In *Regional survey of New York and its environs, Vol. VII:*

Neighborhood and community planning. New York: Regional Plan of New York and Its Environs.

Porta, S., Strano, E., Iacoviello, V., Messora, R., Latora, V., Cardillo, A., . . . Scellato, S. (2009). Street centrality and densities of Retail and Services in Bologna, Italy. *Environment and Planning B: Planning and Design*, *36*(3), 450–465.

Read, S. (1999). Space syntax and the Dutch city. *Environment and Planning B: Planning and Design*, *26*(2), 251–264.

Reps, J. (1965). *The making of urban America: A history of city planning in the United States*. Princeton, NJ: Princeton University Press.

Rosen, K. (2012). *Discrete mathematics and its applications* (7th ed.). New York, NY: McGraw-Hill.

Sadalla, E. K., & Magel, S. G. (1980). The perception of traversed distance. *Environment and Behavior*, *12*(1), 65–79.

Sakellaridi, S., Christova, P., Christopoulos, V. N., Vialard, A., Peponis, J., & Georgopoulos, A. P. (2015). Cognitive mechanisms underlying instructed choice exploration of small city maps. *Frontiers in Neuroscience, 9*. doi:10.3389/fnins.2015.00060

Scoppa, M., & Peponis, J. (2015). Distributed attraction: The effects of street network connectivity upon the distribution of retail frontage in the City of Buenos Aires. *Environment and Planning B: Planning and Design*, *42*(2), 354–378.

Serra, M., Gil, J., & Pinho, P. (2016). Towards an understanding of morphogenesis in metropolitan street-networks. *Environment and Planning B: Urban Analytics and City Science*. doi:10.1177/0265813516684136

Sevtsuk, A. (2010). *Path and place: A study of urban geometry and retail activity in Cambridge and Somerville, MA.* (Doctoral dissertation), Massachusetts Institute of Technology, MA, USA.

Siksna, A. (1998). City centre blocks and their evolution: A comparative study of eight American and Australian CBDs. *Journal of Urban Design*, *3*(3), 253–283.

Smith, M. E. (2007). Form and meaning in the earliest cities: A new approach to ancient urban planning. *Journal of Planning History*, *6*(1), 3–47.

Snellen, D., Borgers, A., & Timmermans, H. (2002). Urban form, road network type, and mode choice for frequently conducted activities: A multilevel analysis using quasi-experimental design data. *Environment and Planning A: Economy and Space*, *34*(7), 1207-1220. doi:10.1068/a349

Southworth, M., & Owens, P. M. (1993). The evolving metropolis: Studies of community, neighborhood, and street form at the urban edge. *Journal of the American Planning Association, 59*(3), 271–287.

Stein, C. S. (1957). *Toward new towns for America*. New York, USA: Reinhold Publishing Corporation.

Strano, E., Viana, M., da Fontoura Costa, L., Cardillo, A., Porta, S., & Latora, V. (2013). Urban street networks, a comparative analysis of ten European cities. *Environment and Planning B: Planning and Design, 40*(6), 1071-1086. doi:10.1068/b38216

Turner, A., Penn, A., & Hillier, B. (2005). An algorithmic definition of the axial map. *Environment and Planning B: Planning and Design, 32*(3), 425–444.

Unwin, R. (1994). *Town planning in practice: An introduction to the art of designing cities and suburbs*. New York, NY: Princeton Architectural Press.

Xie, F., & Levinson, D. (2007). Measuring the structure of road networks. *Geographical Analysis, 39*(3), 336-356. doi:10.1111/j.1538-4632.2007.00707.x